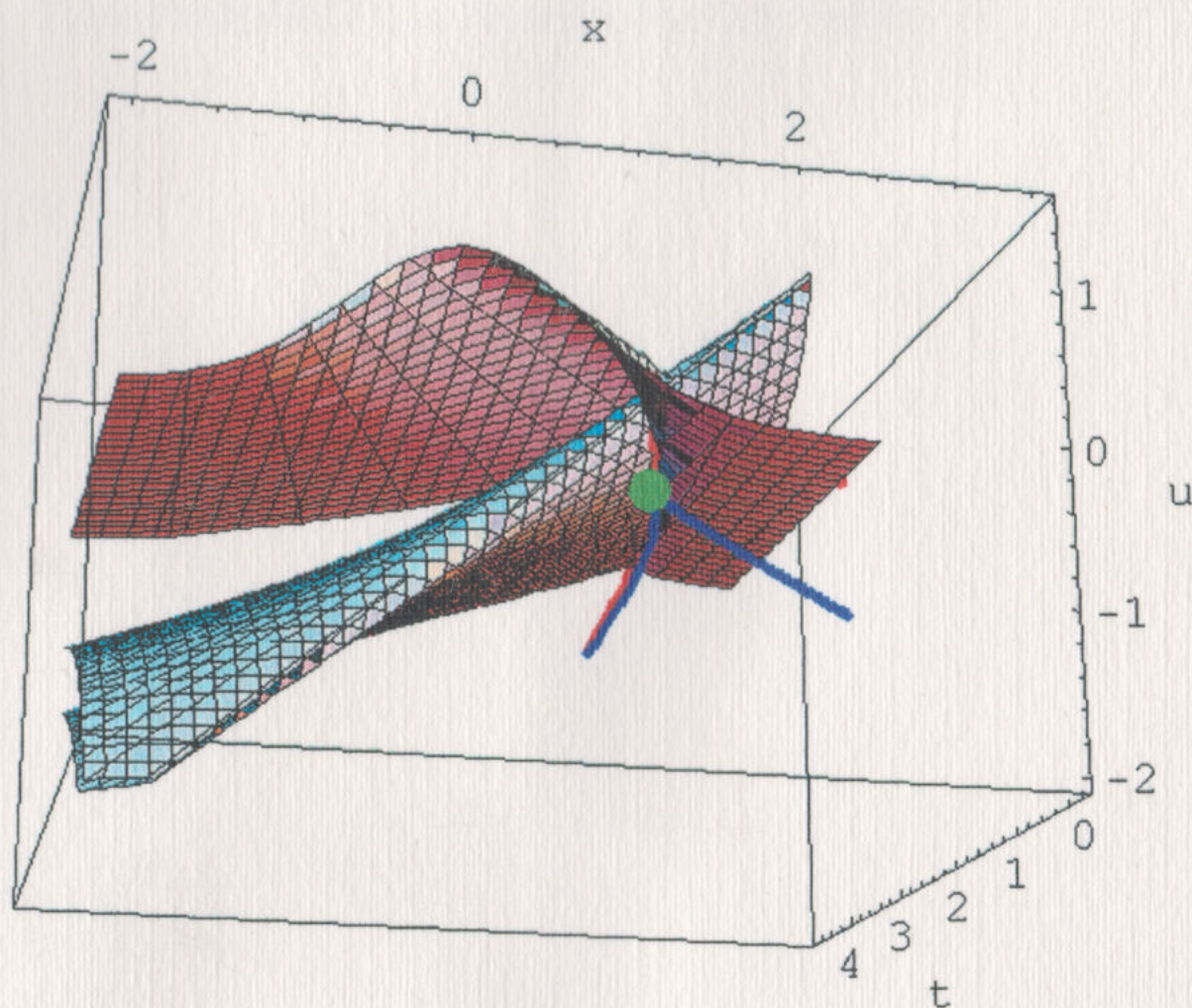


Werner Sanns

Catastrophe Theory with Mathematica

A Geometric Approach



DER ANDERE VERLAG

Di 14.6.05

Catastrophe Theory with Mathematica

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Sanns, Werner:

Catastrophe theory with Mathematica : a geometric approach / Werner Sanns.-

Osnabrück : Der Andere Verlag, 2000

ISBN 3-934366-76-7

© Copyright 2000 by Der Andere Verlag, 49074 Osnabrück

Titelgestaltung: aaron

Der Andere Verlag, Kollegienwall 22A, 49074 Osnabrück

Tel. (0541) 20 20 125, Fax (0541) 20 20 127

E-Mail: talkto@der-andere-verlag.de; dav@skydsl.de

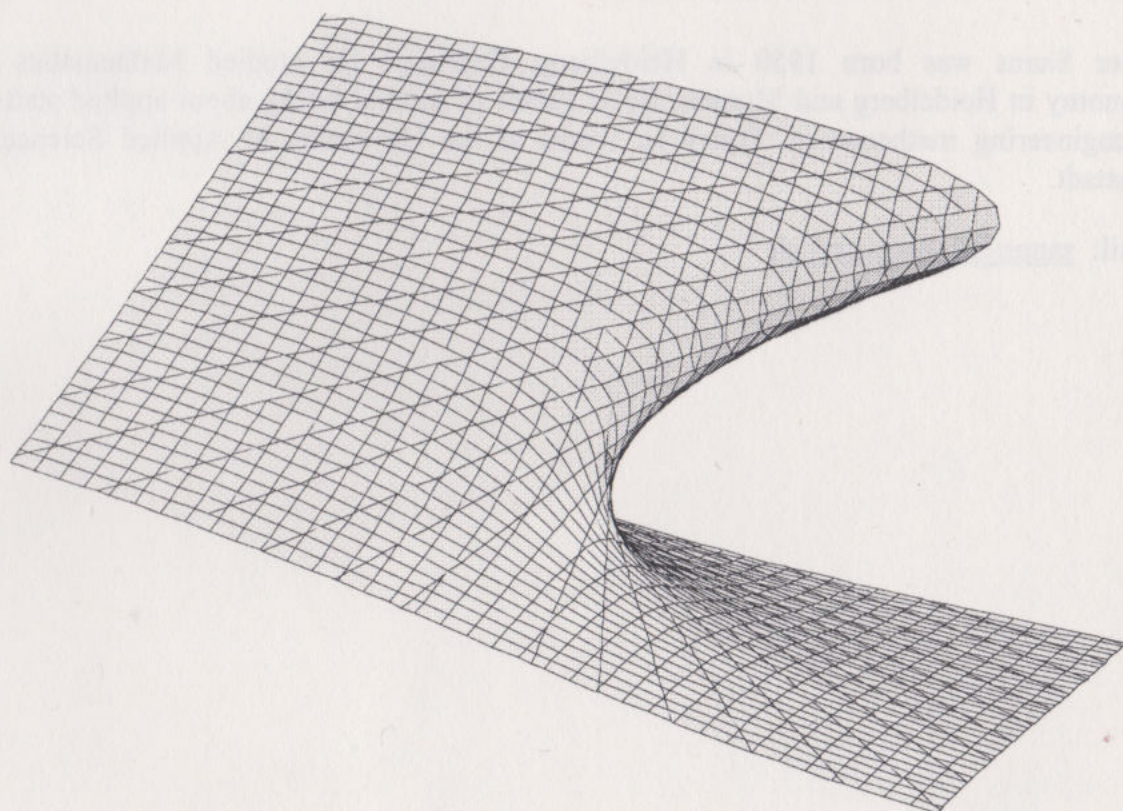
Internet: <http://www.der-andere-verlag.de>

ISBN 3-934366-76-7

Werner Sanns

Catastrophe Theory with Mathematica

A Geometric Approach





Werner Sanns was born 1950 in Heidelberg, Germany. He studied Mathematics and Astronomy in Heidelberg and Munich. He is author of several books about applied statistics and engineering mathematics. Today he works at the University of Applied Sciences in Darmstadt.

e-mail: sanns@fh-darmstadt.de

To my wife

Angelika

and my children

Alexander, Stefan and Christine

To my wife

Angelika

and my children

Alexander, Stefan and Christine

Contents

<u>1</u>	<u>DIFFERENTIAL GEOMETRY AND MATHEMATICA</u>	<u>13</u>
1.1	CURVES AND TANGENTS	13
1.2	SURFACES AND TANGENT PLANES	23
1.3	SMOOTH MAPS	26
1.4	DIFFEOMORPHISMS	31
<u>2</u>	<u>GERMS OF FUNCTIONS</u>	<u>35</u>
<u>3</u>	<u>JETS</u>	<u>39</u>
<u>4</u>	<u>R-EQUIVALENCE</u>	<u>43</u>
<u>5</u>	<u>THE LOCAL BEHAVIOR OF SMOOTH MAPS</u>	<u>48</u>
<u>6</u>	<u>UNFOLDINGS</u>	<u>50</u>
<u>7</u>	<u>THE SEVEN ELEMENTARY CATASTROPHES</u>	<u>57</u>
7.1	THE FOLD	62
7.2	THE CUSP	67
7.3	THE SWALLOWTAIL	92
7.4	THE BUTTERFLY	106
7.5	THE HYPERBOLIC UMBILIC	114
7.6	THE ELLIPTICAL UMBILIC	120
7.7	THE PARABOLIC UMBILIC	123
<u>8</u>	<u>SHOCK WAVES AND CATASTROPHE THEORY</u>	<u>126</u>

9	<u>SHORT HISTORY OF CATASTROPHE THEORY</u>	148
----------	---	------------

10	<u>APPENDIX: INTRODUCTION TO MATHEMATICA</u>	149
-----------	---	------------

11	<u>INDEX</u>	167
-----------	---------------------	------------

12	<u>REFERENCES</u>	174
-----------	--------------------------	------------

Preface

This book is intended as a geometric introduction to catastrophe theory, using Mathematica, one of the most powerful modern computer algebra programs. What is the contents of catastrophe theory? One possible answer to this question could be in one sentence:

Catastrophe theory is the mathematical modeling of sudden changes, so called “catastrophes”, in the behavior of natural systems, which may appear in spite of continuous changes of the system parameters.

We are looking at natural systems (biological, physical, chemical, social, political etc.) and try to explain the occurrence of sudden changes in their behavior. The catastrophes surprisingly arise as consequences of continuous changes of variables (parameters), which influence the system.

It is reasonable to distinguish two major ways into this theory: You can approach catastrophe theory from the point of view of

- a) differentiable maps
- b) dynamical systems.

In the first case, the theory is developed in the mathematical language of maps and functions. Here we are interested for those points in the domain of differentiable (potential) functions, where their gradient vanishes. We speak of “singularities of differentiable functions”. Such singularities characterize the stability points of certain systems.

In the second case, one tries to describe the catastrophes in the system’s behavior by means of vector fields, or equivalently, by systems of differential equations. In this book we choose the first approach.

There are many books on catastrophe theory, but these books do not use Mathematica, to declare the geometric aspects in detail. This book is an elementary approach to that beautiful theory in such a way, that the reader will find many definitions and theorems of catastrophe theory, each followed immediately by examples calculated with Mathematica. Thus step by step he becomes well acquainted with the differential geometric foundations of catastrophe theory and its visualization. Once he has mastered this book, the reader can understand more easily the more theoretical books on this subject.

To be able to work with the book and its examples, you need a PC with an installed version of Mathematica (I have used the actual version 4.0, but because we do our input in a standard language, the programs will most probably run in the next versions too). Please note: In this book all input lines, that you should give in, are printed in bold letters. The corresponding answers of the Mathematica system will be printed in normal letters in the lines immediately following the input. The reader is advised not only to read the program lines, but to do all inputs himself, in order to learn from it and to be able to slightly modify the examples. You may learn the foundations of Mathematica in the appendix of this book, as far as you need it for the calculations and the graphics we produce.

I want to thank Mr. Marco Schuchmann, University of Applied Sciences, Darmstadt, for many discussions on this fascinating part of mathematics.

Werner Sanns

This book is intended as a graduate introduction to catastrophe theory, using the language of the most powerful modern computer algebra programs. What is the question at hand? One possible answer to this question could be in one sentence:

Catastrophe theory is the systematic study of sudden changes in the behavior of a system in the presence of small perturbations, which may appear in spite of continuous changes of the system parameters.

We are looking at natural systems (biological, physical, chemical, social, political, etc.) and try to explain the occurrence of sudden changes in their behavior. The catastrophe theory is a branch of mathematics which studies changes of variables (parameters), which influence the system.

It is extremely important to emphasize that catastrophe theory is not a theory. You can spend a lifetime studying it, and you will never get to the end of it.

It is a very rich and deep theory, and it is a very beautiful and elegant theory.

In the first part, the theory is developed in the mathematical language of maps and bifurcations. Then we move on to the study of the stability of equilibria (potential functions), which is the most important part of the theory. We spend a lot of time on the study of the stability of equilibria, which is the most important part of the theory.

In the second part, we try to describe the catastrophe in the system's behavior by means of the theory of bifurcations, by means of different equations. In this book we choose the theory of bifurcations.

There are many books on catastrophe theory, but these books do not use bifurcations to describe the geometric aspects in detail. This book is an elementary approach to the beautiful theory in such a way that the reader will find many definitions and concepts of catastrophe theory, which are different from the ones which are used in the literature. This way we hope to describe the catastrophe in the most elegant way, which is the most elegant way to describe the catastrophe. We hope that this book will be useful to the reader who is interested in the theory and its applications.

To be able to work with the book and its examples, you need a PC with an installed version of Mathematica 4.0 or later. The book is written in a way that the reader can follow the program with the help of the computer. The program will show the results of the calculations, which are printed in the book. The corresponding text of the Mathematica program will be printed in the book. The reader is advised not only to read the program but also to follow the input. The reader is advised to read the program first, but to do so in a way that is not too difficult. In order to learn the book, it is better to read the book first, and then to read the program. The reader is advised to read the book first, and then to read the program. The reader is advised to read the book first, and then to read the program.

I want to thank Dr. Hans-Joachim Schmittmann, University of Applied Sciences, Darmstadt, for many discussions on this fascinating part of mathematics.

Wolfram Janz

1 Differential geometry and Mathematica

First, we want to study some elementary differential geometric objects using Mathematica.

1.1 Curves and Tangents

We are introducing the concept of curves in the n -dimensional space $\mathbb{R}^n$:

A curve in $\mathbb{R}^n$ in parametric form is given by a continuous map

$$c : I \rightarrow \mathbb{R}^n$$

$$t \mapsto c(t)$$

where I means an open or closed interval in $\mathbb{R}$, which contains more than one point. I is also called parameter interval.

For each $t \in I$, $c(t)$ is a vector in $\mathbb{R}^n$, the “curve point $c(t)$ ” with components $c_1(t), \dots, c_n(t)$.

We examine a space curve as our first example with Mathematica, that means a curve in $\mathbb{R}^3$, here a helix. The realization of a space curve in Mathematica is done by the “ParametricPlot3D” statement.

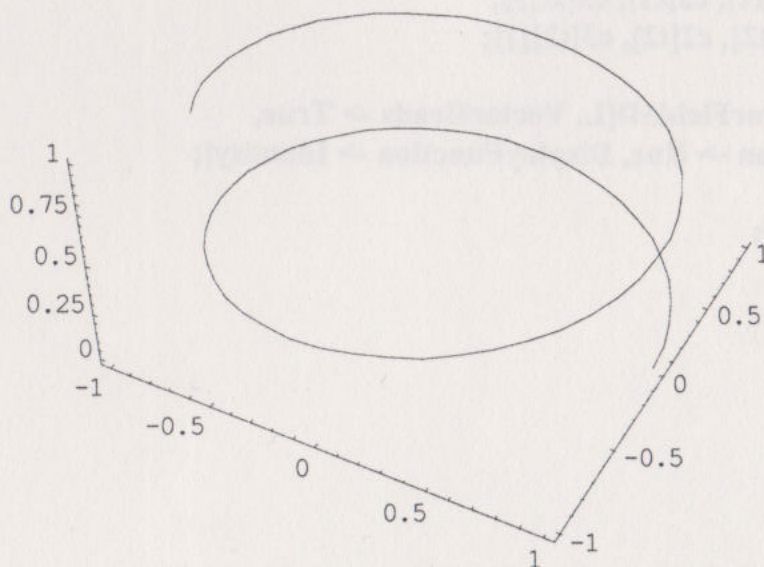
```
c1[t_] := Cos[t];
```

```
c2[t_] := Sin[t];
```

```
c3[t_] := 1/10*t
```

```
c[t_] := {c1[t], c2[t], c3[t]}
```

```
curve1 = ParametricPlot3D[Evaluate[c[t]], {t, 0, 10}, Boxed -> False];
```



A curve is called continuously differentiable, if all its component functions are continuously

differentiable, that means, the derivatives exist and are continuous. The image set $\{c(t)|t \in I\}$ is often also called "the curve c ".

We want to describe now secants and tangents to curves. Let c be a curve and h a real number unequal to zero, such that $t+h \in I$ for $t \in I$, where I is an open interval. The secant between the curve points $c(t)$ and $c(t+h)$ is spanned by the vector $\frac{c(t+h) - c(t)}{h}$. This fact we also want to demonstrate with Mathematica. In the following list L we give the starting point and the components of two vectors. These are graphically generated by the Mathematica statement „ListPlotVectorField3D“. The curve $c(t)$, which we use as our example, is the same as defined above.

Needs["Graphics`PlotField3D`"]

t1 = 1;

t2 = 2;

To the values t_1 and t_2 correspond the curve points $c(t_1)$ and $c(t_2)$:

c[t1]

$\{\text{Cos}[1], \text{Sin}[1], \frac{1}{10}\}$

c[t2]

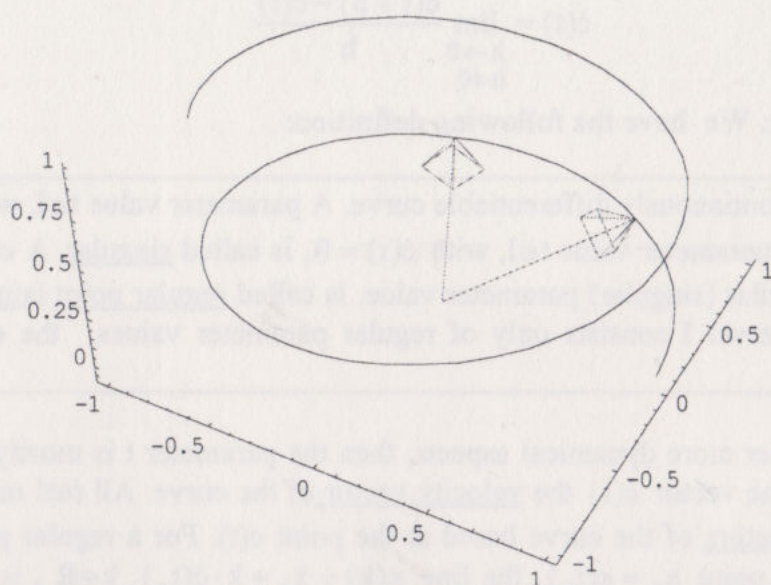
$\{\text{Cos}[2], \text{Sin}[2], \frac{1}{5}\}$

Now we generate the mentioned list L for the vector arrows:

**L = {{{0, 0, 0}, {c1[t1], c2[t1], c3[t1]}},
{{0, 0, 0}, {c1[t2], c2[t2], c3[t2]}}};**

**vcts = ListPlotVectorField3D[L, VectorHeads -> True,
ColorFunction -> Hue, DisplayFunction -> Identity];**

Show[curve1, vcts];

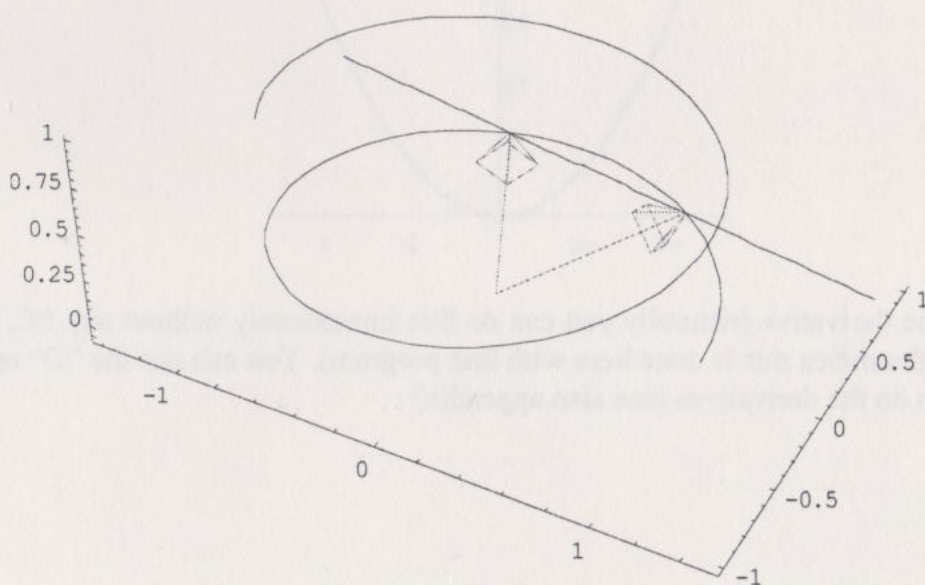


We compute and draw a secant. Here we have chosen as an example $t=t_1=1$ and $t+h=t_2=2$, that is we have $h=1$ (compare also the following computation of the tangent. To the curve point $c(t_1)$ a real multiple (written as k) of the generating vector of the secant is added:

```
Seca[k_]:=c[t1]+k(c[t2]-c[t1])
```

```
Secant=ParametricPlot3D[Evaluate[Seca[k]],  
  {k,-1,2},DisplayFunction->Identity];
```

```
Show[curve1, vcts, Secant,  
  DisplayFunction -> $DisplayFunction];
```



For points $c(t)$ and $c(t+h)$ that „get closer and closer“, that is, for h getting smaller, we have

in the limit the tangent. This is only possible if the generating vector

$$\dot{c}(t) = \lim_{\substack{h \rightarrow 0 \\ h \neq 0}} \frac{c(t+h) - c(t)}{h}$$

is not the zero vector. We have the following definition:

Let $c : I \rightarrow \mathbb{R}^n$ be a continuously differentiable curve. A parameter value $t \in I$, with $\dot{c}(t) \neq 0$, is called regular, and a parameter value $t \in I$, with $\dot{c}(t) = 0$, is called singular. A curve point $c(t)$, that belongs to a regular (singular) parameter value, is called regular point (singular point) of the curve. If the interval I consists only of regular parameter values, the curve is called regular.

If we see curves under more dynamical aspects, then the parameter t is mostly interpreted as time, and one calls the vector $\dot{c}(t)$ the velocity vector of the curve. All real multiples of $\dot{c}(t)$ are called tangent vectors of the curve based at the point $c(t)$. For a regular point t_0 and the corresponding curve point $x_0 = c(t_0)$, the line $x(k) = x_0 + k \cdot \dot{c}(t_0)$, $k \in \mathbb{R}$, is the tangent in x_0 .

Examples:

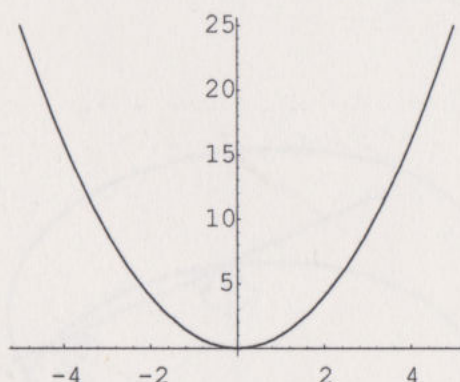
a) The graph of a function $f: \mathbb{R} \supseteq I \rightarrow \mathbb{R}$ can be interpreted as parameter curve

$$c: I \rightarrow \mathbb{R}^2$$

$$t \mapsto c(t) = (c_1(t), c_2(t)) = (t, f(t))$$

Look as an example at the graph of the parabola $f(t)=t^2$, that is, the curve $c(t)=(t, t^2)$

ParametricPlot[{t, t^2}, {t, -5, 5}]



We compute the derivative (naturally you can do this immediately without any PC, but as an exercise in Mathematica this is done here with that program). You can use the “D” operator in Mathematica to do the derivatives (see also appendix) :

c[t_]:= {t,t^2}

D[c[t],t]

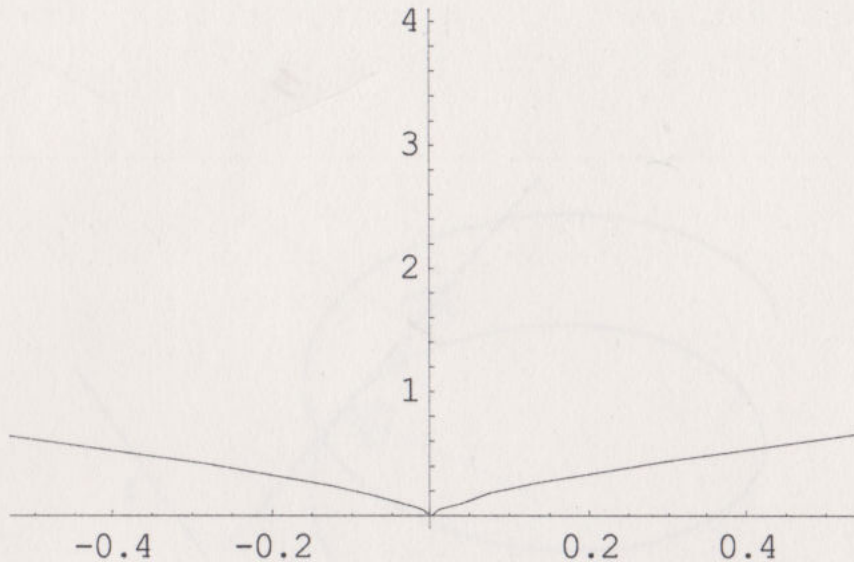
{1,2 t}

We find $\dot{c}(t) = (1, 2t) \neq (0,0) \forall t \in \mathbb{R}$. This means all points of the parabola are regular.

b) Neil's parabola: $c(t) = (t^3, t^2)$

$c[t_] := \{t^3, t^2\}$

`ParametricPlot[Evaluate[c[t]], {t, -2, 2}];`



$D[c[t], t]$

$\{3t^2, 2t\}$

We have $\dot{c}(0) = (0,0)$, which means, that at the origin no tangent can exist, the parameter value 0 is singular. The curve has a cusp there.

The tangent of the helix from the above example will now be drawn for $t_1=1$ in $c(t_1)$ together with a generating vector. You can use parts of the program lines, that you have already given in above.

$c1[t_] := \text{Cos}[t];$

$c2[t_] := \text{Sin}[t];$

$c3[t_] := 1/10*t$

$c[t_] := \{c1[t], c2[t], c3[t]\}$

$NV = \{0,0,0\};$

$L = \{\{NV, c[t_1]\}, \{c[t_1], c'[t_1]\}\};$

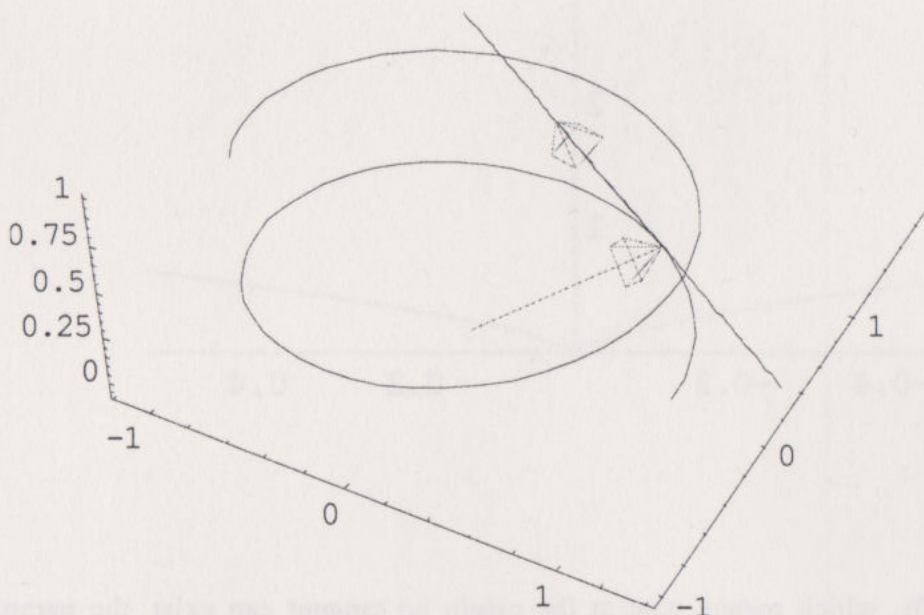
vectors =

`ListPlotVectorField3D[L, VectorHeads -> True,
ColorFunction -> Hue, DisplayFunction -> Identity];`

```
tang[k_] := c[t1] + k*c'[t1]
```

```
tangent=ParametricPlot3D [Evaluate[tang[k]],  
{k,-1,2}, DisplayFunction->Identity];
```

```
Show [curve1, vectors, tangent,  
DisplayFunction -> $DisplayFunction];
```



Quasi as counterpart of a curve $c:I \rightarrow \mathbb{R}^n$, we now examine differentiable functions defined on an open subset U of $\mathbb{R}^n$, that means, real valued differentiable maps $f: \mathbb{R}^n \supseteq U \rightarrow \mathbb{R}$.

If f is such a differentiable function defined on an open neighborhood U of a point $x_0 \in \mathbb{R}^n$, and $c:I \rightarrow U \subseteq \mathbb{R}^n$ is a differentiable curve through $x_0 = c(t_0)$, $t_0 \in I$, then $f \circ c: I \rightarrow \mathbb{R}$ is differentiable at t_0 and by the chain rule we have:

$$(f \circ c)'(t_0) = \sum_{i=1}^n \frac{\partial f}{\partial x_i}(x_0) \cdot \frac{dc_i}{dt}(t_0)$$

This is the dot product of the gradient

$$\text{grad } f(x_0) = \left(\frac{\partial f}{\partial x_1}(x_0), \dots, \frac{\partial f}{\partial x_n}(x_0) \right)$$

of the function f in x_0 with the tangent vector

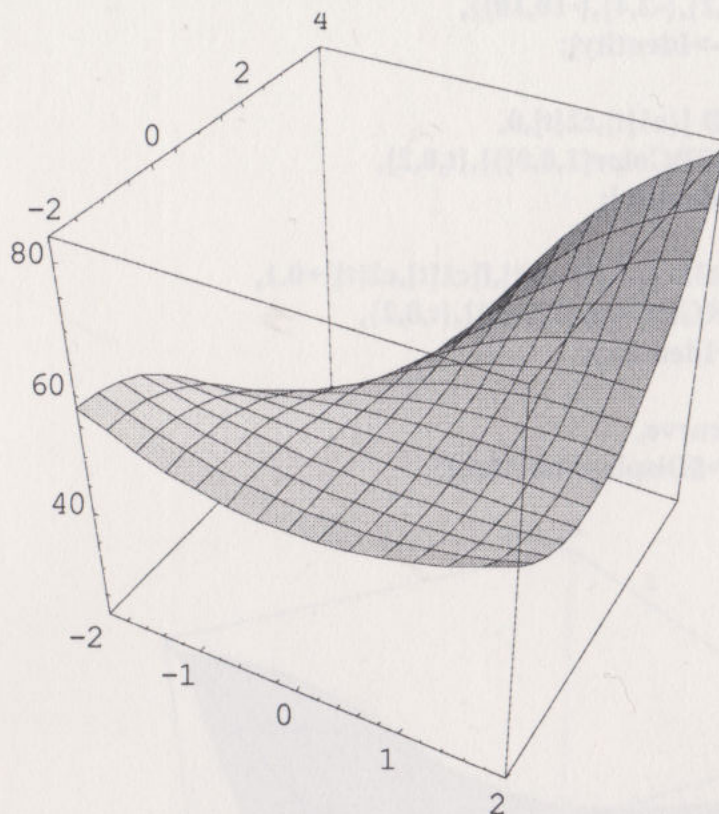
$$\dot{c}(t_0) = \left(\frac{dc_1}{dt}(t_0), \dots, \frac{dc_n}{dt}(t_0) \right).$$

$(f \circ c)'$ is called the derivative of f in the direction of c . Let's calculate an example with Mathematica. We define a function f in two variables x and y and draw its graph.

```
f[x_,y_] := -x^2 * y + y^2 * x + 2x * y + 50
```



```
F=Plot3D[f[x,y],{x,-2,2},{y,-2,4},BoxRatios->{1,1,1}];
```



We choose a point (x_0, y_0) , where we want to determine the directional derivative. The curve in parameter space be defined by $c(t)=(t, t^2)$:

```
x0=1;
```

```
y0=1;
```

```
{c1[t_],c2[t_]}={t,t^2};
```

```
c[t_]:= {c1[t],c2[t]}
```

```
f[c1[t],c2[t]]
```

```
50 + 2 t^3 - t^4 + t^5
```

Later we want to show the parameter plane and the graph of f (surface) in the same picture. So we define:

```
e[x_,y_]:=0
```

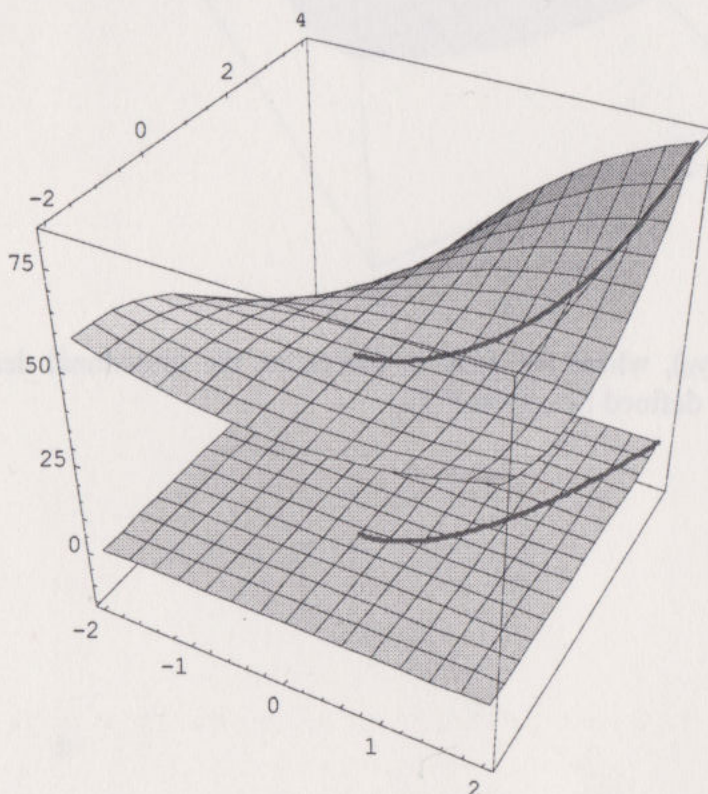
Now we generate three graphics, one for the plane, one for the curve $c(t)$ laying in that plane, and one for the space curve, being above $c(t)$ on the surface. The graphics then are shown together with the help of the "Show" command. If you want to see each graphic alone, please drop the option "DisplayFunction->Identity" and the colon in front of it in the following input lines.

```
plane=Plot3D[e[x,y],{x,-2,2},{y,-2,4},
  PlotRange->{{-2,2},{-2,4},{-10,10}},
  DisplayFunction->Identity];
```

```
curve=ParametricPlot3D[{c1[t],c2[t],0,
  {Thickness[0.008],RGBColor[1,0,0]}},{t,0,2},
  DisplayFunction->Identity];
```

```
spcurve=ParametricPlot3D[{c1[t],c2[t],f[c1[t],c2[t]]+0.1,
  {Thickness[0.008],RGBColor[1,0,0]}},{t,0,2},
  DisplayFunction->Identity];
```

```
Show[F,plane,curve, spcurve,
  DisplayFunction->$DisplayFunction];
```



We compute $c'(t)$ (Mathematica uses ' instead of dot) and insert $t=1$, because this leads to the point $x_0=1, y_0=1$:

```
c'[t]
```

```
{1,2 t}
```

```
v=c'[t]/. t->1
```

```
{1,2}
```

Now we must determine gradient f . This can be done in Mathematica by the "Outer" statement. It performs an application of the derivative Operator D to all functions by all

variables. The functions and variables in the definition are of type "List". See also part 1.3 on differentiable functions.

Jacobi[functions_List, variables_List] := Outer[D, functions, variables]

gradf=Jacobi[{f[x,y]},{x,y]}

{ {2 y - 2 x y + y², 2 x - x² + 2 x y} }

The directional derivative at (x_0, y_0) is given as dot product (in Mathematica ".") of the gradient with the tangent vector v to c and inserting (x_0, y_0) :

ra=gradf.v /.{x->x0,y->y0}

{7}

This value is needed as the third component of a space vector, whose first and second components are given by v .

vr=Flatten[Append[v,ra]]

{1,2,7}

With these values we compute the equation of the tangent at the point $(x_0, y_0, f(x_0, y_0))$. Here we write t for the parameter in the equation of the tangent:

tang[t_]:={x0,y0,f[x0, y0]}+t*vr

tang[t]

{1+t,1+2 t,52+7 t}

We now generate a picture for the tangent, which is not yet shown on the PC display (option "DisplayFunction->Identity").

**tg=ParametricPlot3D [{1+t,1+ 2t,52+7 t+0.5, {Thickness[0.005],RGBColor[0,0,1]}},
{t, -1, 2},DisplayFunction->Identity];**

The projection onto the plane $e(x,y)=0$ of the tangent to the space curve gives the tangent to the curve in this plane.

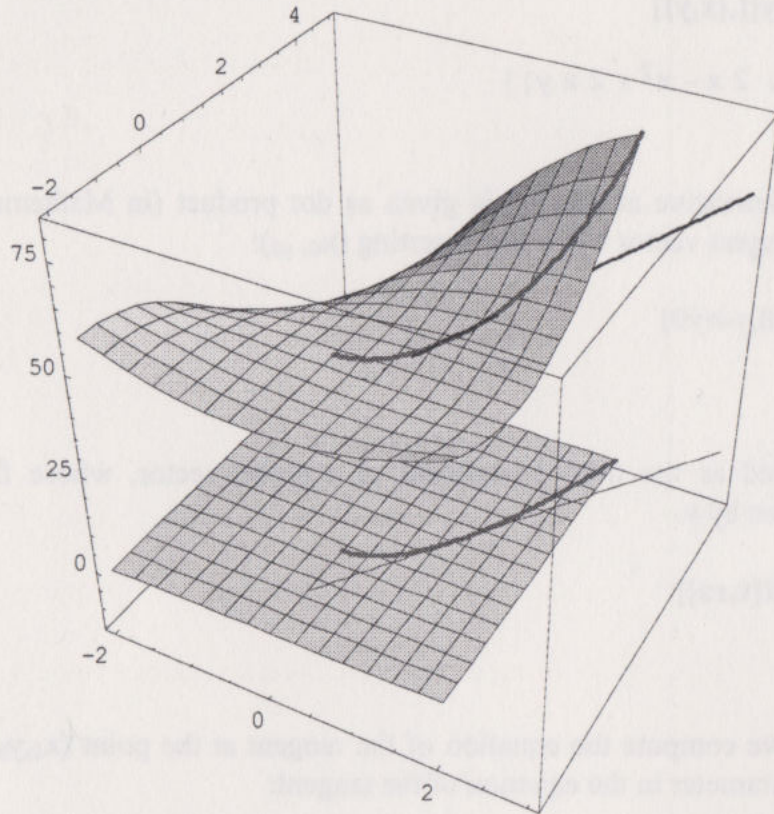
planetang[t_]={1+t,1+2t,0}

{1+t,1+2 t,0}

**ptg=ParametricPlot3D[Evaluate[planetang[t]],
{t, -1, 2},DisplayFunction->Identity];**

We now present all objects in a single graphic:

**Show[F,plane,curve, spcurve,tg,ptg,
DisplayFunction->\$DisplayFunction];**



1.2 Surfaces and tangent planes

Again we start with a definition:

A parameterized surface patch in $\mathbb{R}^3$ is given by a bijective, continuously differentiable map

$$X: G \rightarrow \mathbb{R}^3$$

$$(u,v) \mapsto X(u,v)$$

where G is some bounded region in the parameter plane $\mathbb{R}^2$, and the vectors $\frac{\partial X}{\partial u}$ and $\frac{\partial X}{\partial v}$ are linearly independent.

Note that:

1) The fact that $\frac{\partial X}{\partial u}$ and $\frac{\partial X}{\partial v}$ are linearly independent is often formulated as $\frac{\partial X}{\partial u} \times \frac{\partial X}{\partial v} \neq 0$.

It means, that the tangent plane in each point of the surface patch exists. The surface patch is said to be “smooth”.

2) Parameterized surfaces patches are generalized by the concept of surfaces. A surface F is a connected point set in $\mathbb{R}^3$, where, for each of its points, there exist an open ball K , containing that point, such that $F \cap K$ consists of an at most countable infinite number of parameterized surface patches. Surfaces and curves again are special cases of so called manifolds (s. chap.: diffeomorphisms). We often designate parametric surfaces patches simply as “surfaces”.

We look at the following example with Mathematica:

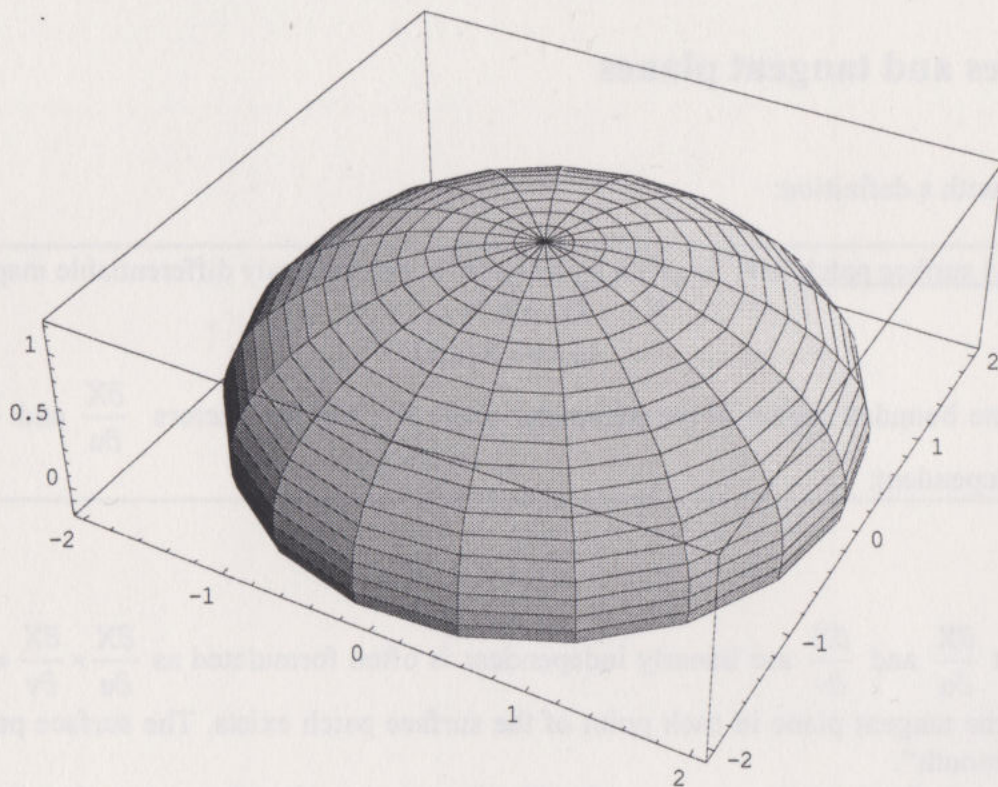
```
Remove["Global`*"]
```

```
a=2;
```

```
b=1.2;
```

```
X[u_,v_]:={a Cos[u] Sin[v],a Sin[u] Sin[v],b Cos[v]}
```

```
M=ParametricPlot3D [X[u,v]//Evaluate,{u,0,2 Pi},{v,0,0.55*Pi}]
```



We now want to draw a point on the surface together with the tangent plane. Let $(u_0, v_0) = (0.5, 0.9)$ be a point in parameter space and p_0 be its image.

```
u0=0.5;
```

```
v0=0.9;
```

```
p0=X[u0,v0]
```

```
{1.37487 , 0.751094 , 0.745932 }
```

A graphic is generated for that point (which is not yet to be seen on the PC display):

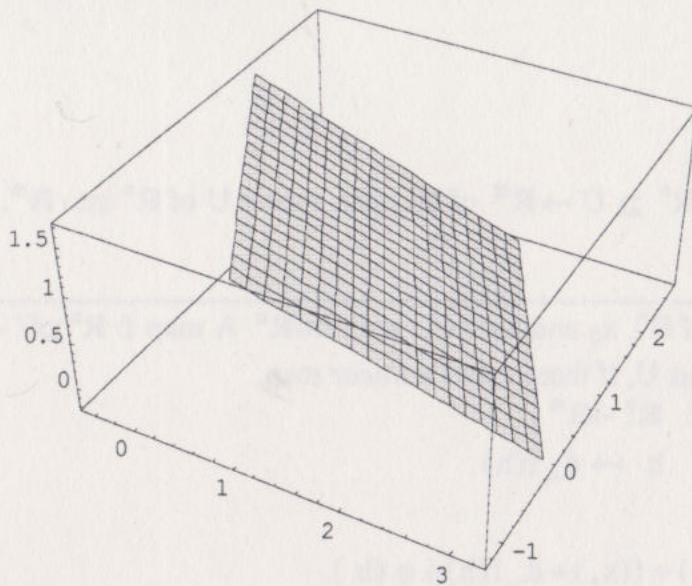
```
bp=Graphics3D[{RGBColor[1,0,0],PointSize[0.01],Point[p0]}];
```

Now the tangent plane is calculated as sum of p_0 and a linear combination of the tangent vectors $\frac{\partial X}{\partial u}$ and $\frac{\partial X}{\partial v}$:

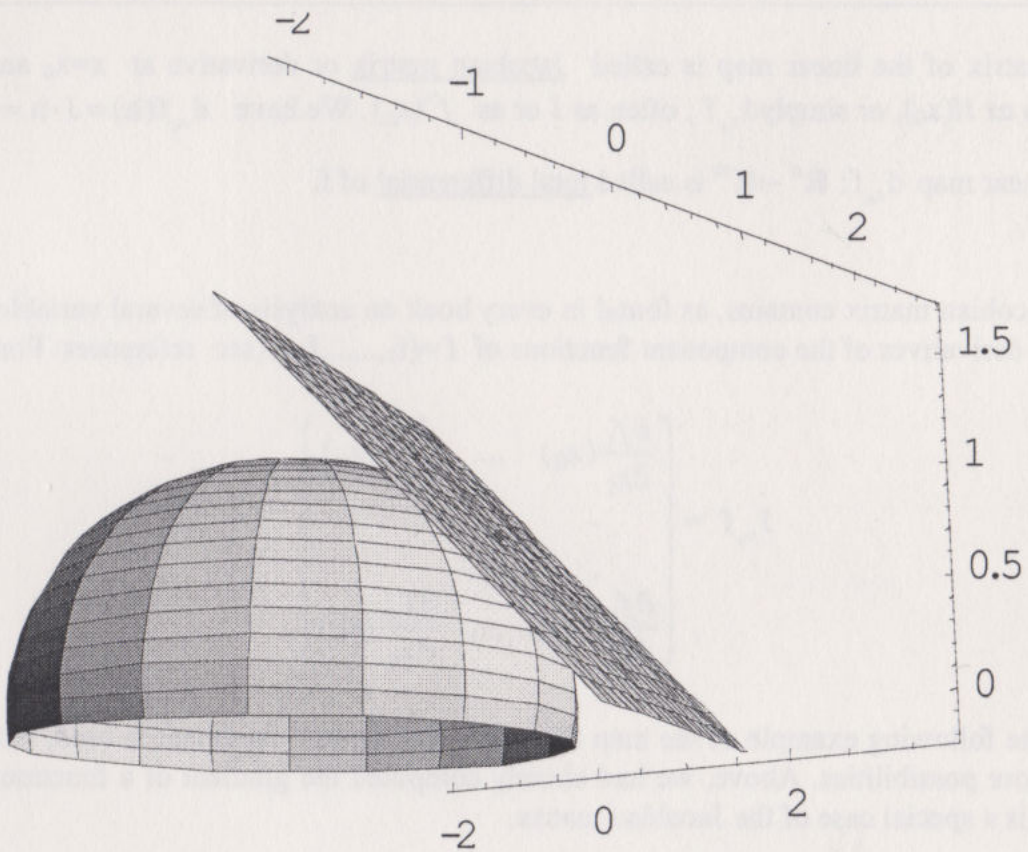
```
t=p0+k1 D[X[u,v],u]+k2 D[X[u,v],v]/.{u->u0,v->v0}
```

```
{1.37487 - 0.751094 k1 + 1.09103 k2, 0.751094 + 1.37487 k1 + 0.596031 k2, 0.745932 - 0.939992 k2}
```

```
tr=ParametricPlot3D [t/Evaluate,{k1,-1,1},{k2,-1,1}];
```

Show[M,bp,tr,ViewPoint->{1.5,-1,-0.3},Boxed->False];



1.3 Smooth maps

Now we consider maps $f: \mathbb{R}^n \supseteq U \rightarrow \mathbb{R}^m$ of an open subset U of $\mathbb{R}^n$ into $\mathbb{R}^m$. The following definition is well known:

Let U be an open subset of $\mathbb{R}^n$, x_0 and $x_0+h \in U$ with $h \in \mathbb{R}^n$. A map $f: \mathbb{R}^n \supseteq U \rightarrow \mathbb{R}^m$ is called (totally) differentiable in $x_0 \in U$, if there exists a *linear* map

$$\begin{aligned} d_{x_0} f: \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ h &\mapsto d_{x_0} f(h) \end{aligned}$$

such that:

$$f(x_0 + h) = f(x_0) + d_{x_0} f(h) + \varphi(h)$$

with a map $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^m$, defined in some neighborhood of $0 \in \mathbb{R}^n$, for which $\lim_{h \rightarrow 0} \frac{\varphi(h)}{|h|} = 0$.

Here $|h|$ is the Euclidian norm of h .

The matrix of the linear map is called Jacobian matrix or derivative at $x=x_0$ and is often written as $Jf(x_0)$, or simply $J_{x_0} f$, often as J or as $f'(x_0)$. We have $d_{x_0} f(h) = J \cdot h = f'(x_0) \cdot h$.

The linear map $d_{x_0} f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is called total differential of f .

The Jacobian matrix contains, as found in every book on analysis of several variables, just the partial derivatives of the component functions of $f=(f_1, \dots, f_m)$ (see references: Forster).

$$J_{x_0} f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_0) & \dots & \frac{\partial f_1}{\partial x_n}(x_0) \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(x_0) & \dots & \frac{\partial f_m}{\partial x_n}(x_0) \end{pmatrix}$$

In the following example of the map $f: \mathbb{R}^2 \rightarrow \mathbb{R}^3$, the “Outer” statement is used, but there are still more possibilities. Above, we had already computed the gradient of a function this way, which is a special case of the Jacobian matrix.

Remove["Global`*"]

f[x_,y_]:= {x^2 y, 2 y^2 x^3, x y}

Jacobi[functions_List, variables_List] := Outer[D, functions, variables]

Jacobi[f[x,y],{x,y}]/MatrixForm

$$\begin{pmatrix} 2xy & x^2 \\ 6x^2y^2 & 4x^3y \\ y & x \end{pmatrix}$$

In Catastrophe theory one tries to characterize the behavior of systems by differentiable maps $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$. By a differentiable function we mean a differentiable map in the special case $m=1$.

For the points in the domain of a differentiable map, we make a distinction between regular and singular points:

A differentiable map $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, defined in some neighborhood of the point $x_0 \in \mathbb{R}^n$, is called singular in x_0 , if

$$\text{rank } J_{x_0} f < \min \{n, m\}.$$

Otherwise f is called regular in x_0 .

Here again $J_{x_0} f$ is the Jacobian matrix of f at $x=x_0$. For $m=1$, it is, for a differentiable function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, this is formulated as follows:

A differentiable function $f: (\mathbb{R}^n, x_0) \rightarrow \mathbb{R}$ is called singular in $x_0 \in \mathbb{R}^n$, if

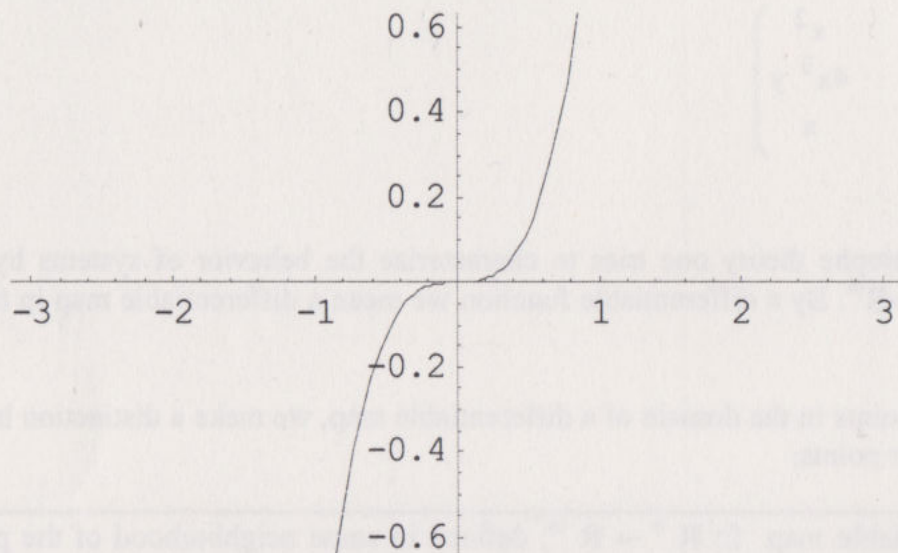
$$\text{grad } f(x_0) = 0$$

The notation $f: (\mathbb{R}^n, x_0) \rightarrow \mathbb{R}$ should make clear, that f is defined in some neighborhood of x_0 . The 0 on the right side of $\text{grad } f(x_0) = 0$ means the zero vector, but is often written simply as 0. The point x_0 , where the function is singular, is called singularity of the function. Often the name “singularity” stands for the function itself, if it is singular at one point x_0 (mostly the origin of $\mathbb{R}^n$). A point, where a function is singular, is also called critical point. A critical point of a function f is called degenerated critical point of that function, if the Hessian of the second order partial derivatives (a symmetric quadratic matrix) is singular there, it is, its determinant is zero there. We show some examples for singularities (here singular at $x=0$):

First, the function $g(x)=x^3$:

`g[x_]:=x^3`

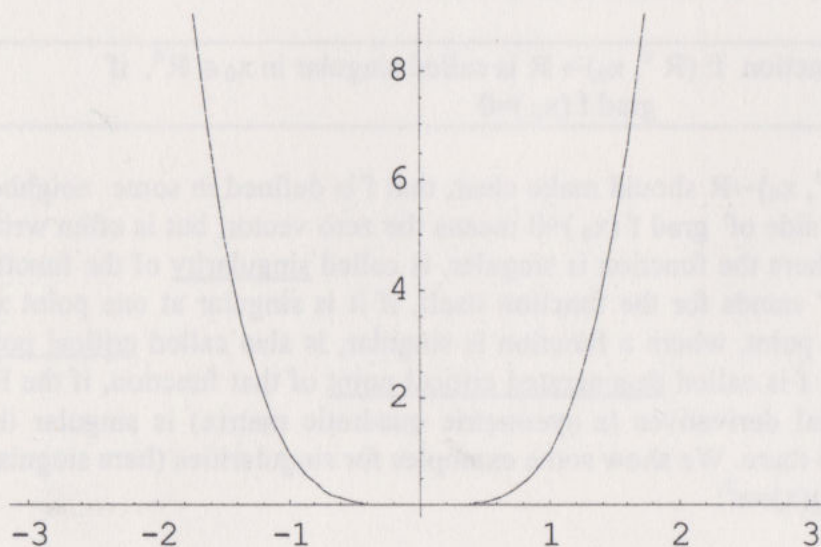
`Plot[g[x],{x,-3,3}];`



The second example is the function x^4 :

```
h[x_]:=x^4
```

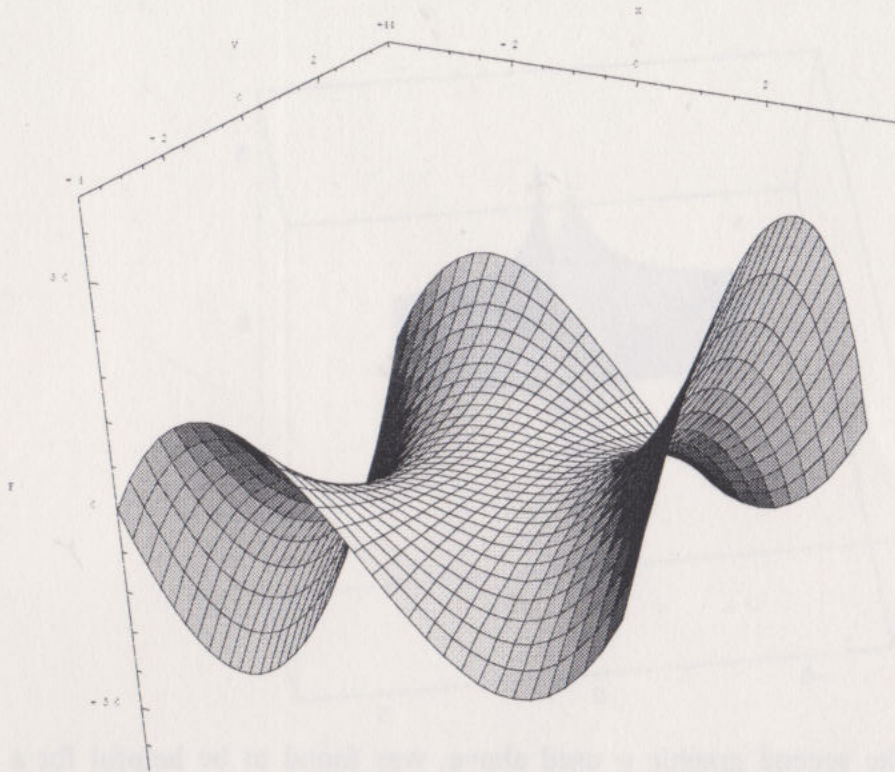
```
Plot[h[x],{x,-3,3}];
```



An example for a singularity in two variables is the following function $x^3 - x \cdot y^2$ (the so called monkey-saddle):

```
g[x_,y_]:=x^3 - x y^2
```

```
Plot3D[g[x,y],{x,-4,4},{y,-4,4}, AspectRatio->2, BoxRatios->{1,1,2},
Boxed->False,PlotPoints->30,ViewPoint->{1.3,-2.4,2.},Mesh->True,
AxesLabel->{"x","y","F"},ColorOutput->CMYKColor];
```

The reason, why we are interested in singularities, is as follows. Many systems can be described by potential functions. These are differentiable functions, whose negative gradient characterizes the forces acting in the system at each place. If these forces are in balance, that is, their sum equals the zero vector, then the gradient of the potential is the zero vector, the potential function is singular there. But Catastrophe theory doesn't look merely at single potential functions, but rather at whole families of (parametric) functions:

Catastrophe theory works with families of potential functions, which control the system's behavior.

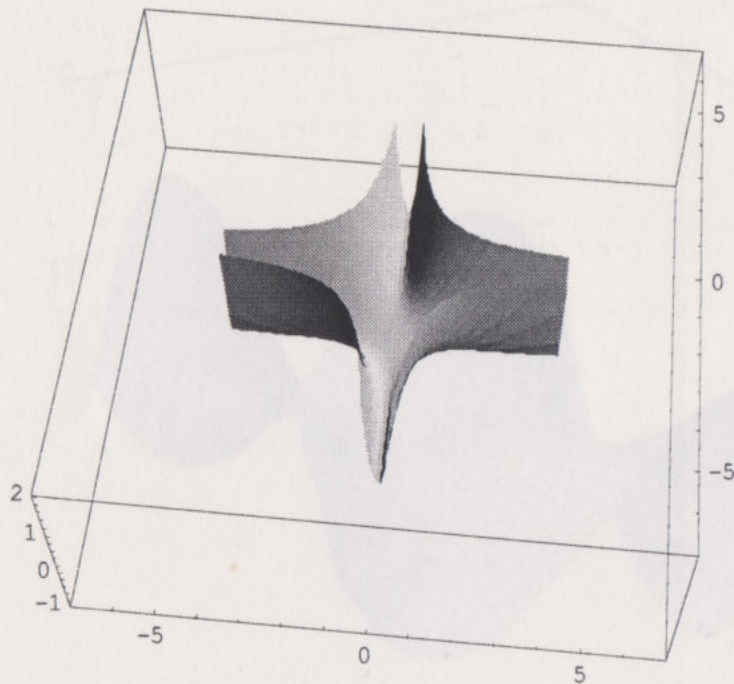
For a function in one variable, the vanishing of the first derivative means the existence of an extremum or a saddle point. For functions in several variables, there is a much larger variety for such points as only extrema and saddle points, as the following example of the "crossed pig trough" shows:

```
trog[x_,y_]:=x^2 * y^2
```

```
tr=Plot3D[trog[x,y],{x,-4,4},{y,-4,4}, PlotRange->{{-7,7},{-7,7},{-1,2}},PlotPoints->70,
BoxRatios->{1,1,0.4},ClipFill->None,ViewPoint->{0.175, -1.752, 2.707},
Mesh->False, DisplayFunction->Identity];
```

```
p=ParametricPlot3D[{0,0,0},{t,0,1},DisplayFunction->Identity];
```

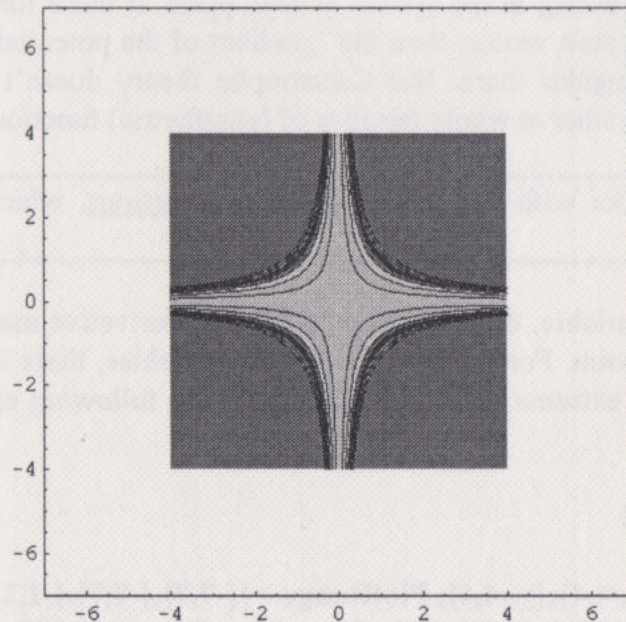
```
Show[tr,p,DisplayFunction->$DisplayFunction]
```



Technical hint: The second graphic p used above, was found to be helpful for a better smoothing of the borders of the surface.

The contour lines of this function are drawn by the “ContourGraphics” statement:

**Show[ContourGraphics[tr], ColorFunction -> Hue,
DisplayFunction -> \$DisplayFunction];**



The origin is not an isolated singular point, that means, in each neighborhood of that point, there is still another singular point.

1.4 Diffeomorphisms

If we are interested in the structure of the singularities of differentiable maps, we must know, which coordinate transformations we may admit, to get qualitatively equal singularity sets. This leads to “diffeomorphisms”.

A diffeomorphism is a bijective differentiable map between open sets of $\mathbb{R}^n$, whose inverse is differentiable too.

As an example, we take the following map of the positive first quadrant of the plane into itself:

$$f: \mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2$$

$$f(x,y) = (x^2 \cdot y, x \cdot y) = (u,v).$$

With Mathematica we may declare this as follows:

```
Remove["Global`*"]
```

```
<<Graphics`Master`
```

```
u:=x^2 y;
```

```
v:=x y;
```

```
f[x_,y_]={u,v}
```

```
{ x^2 y, x y }
```

To calculate its Jacobian matrix Jf, we write as input lines:

```
Jacobi[functions_List, variables_List] := Outer[D, functions, variables]
```

```
Jf=Jacobi[{u,v},{x,y}]/MatrixForm
```

$$\begin{pmatrix} 2xy & x^2 \\ y & x \end{pmatrix}$$

```
Det[%]
```

$$x^2 y$$

Its determinant is not zero at each point of the domain of f (positive first quadrant). We calculate the inverse map:

```
Clear[u,v]
```

```
Invmap=Solve[{u==x^2 y,v==x y},{x,y}]
```

$$\left\{ \left\{ y \rightarrow \frac{v^2}{u}, x \rightarrow \frac{u}{v} \right\} \right\}$$

To demonstrate the effect of f , we at first draw a region (here a circle) and some coordinate lines, which we then map by f .

```
g1=ImplicitPlot[(x-2)^2+(y-2)^2== 1, {x, -2, 5}, PlotRange->{{-1,5}, {-1,5}},
  PlotPoints->100,Prolog->{Thickness[0.007],RGBColor[1,0,1]},
  DisplayFunction->Identity];
```

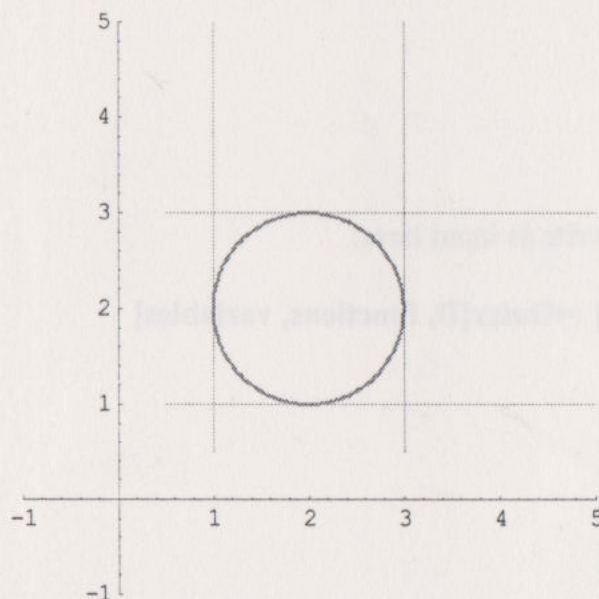
```
Coordlin1=Graphics[{Thickness [0.005],RGBColor[1,0,0],
  Line[{{0.5,1},{6,1}}]},Axes->False];
```

```
Coordlin2=Graphics[{Thickness [0.005],RGBColor[0.1,0.5,0.1],
  Line[{{0.5,3},{6,3}}]},Axes->False];
```

```
Coordlin3=Graphics[{Thickness [0.005],RGBColor[0,0,1],
  Line[{{1,0.5},{1,6}}]},Axes->False];
```

```
Coordlin4=Graphics[{Thickness [0.005],RGBColor[0,0.3,0.4],
  Line[{{3,0.5},{3,6}}]},Axes->False];
```

```
Show[g1,Coordlin1, Coordlin2,Coordlin3,Coordlin4,
  DisplayFunction->$DisplayFunction];
```



We express x and y by the coordinates u and v :

```
{x,y}={x,y} /. Invmapping // Flatten
```

$$\left\{ \frac{u}{v}, \frac{v^2}{u} \right\}$$

x

$$\frac{u}{v}$$

y

$$\frac{v^2}{u}$$

Now we can draw the mapped objects, i.e. the images of the circle and coordinate lines.

```
g2=ImplicitPlot[(x-2)^2+(y-2)^2==1,{u,0.1,25},{v,0.1,10},
  PlotPoints->100,PlotStyle->{Thickness[0.004],
  RGBColor[1,0,1]},DisplayFunction->Identity];
```

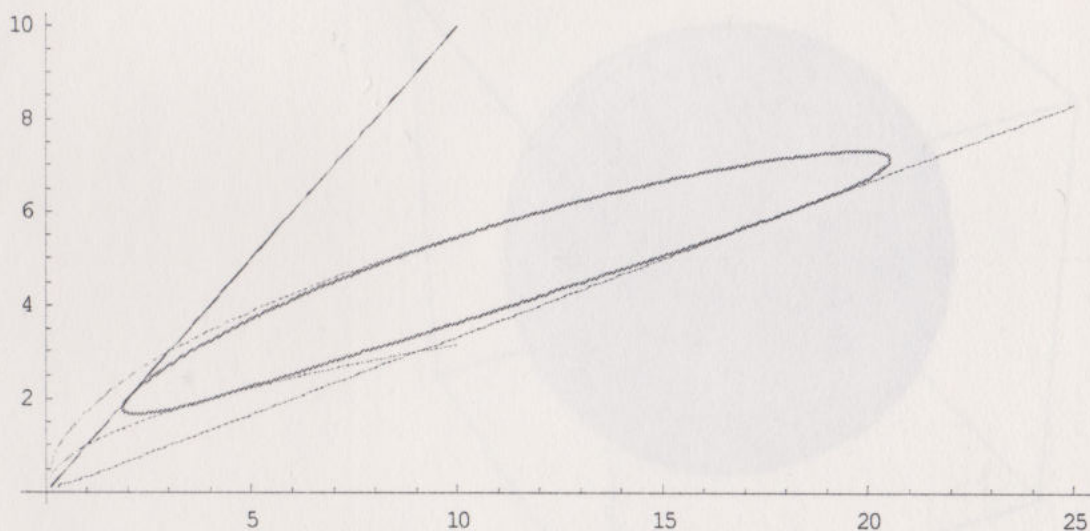
```
g3=ImplicitPlot[y==1,{u,0.1,10},{v,0.1,10},
  PlotStyle->RGBColor[1,0,0],
  PlotPoints->100,DisplayFunction->Identity];
```

```
g4=ImplicitPlot[y==3,{u,0.1,10},{v,0.1,10},
  PlotStyle->RGBColor[0.1,5,1],
  PlotPoints->100,DisplayFunction->Identity];
```

```
g5=ImplicitPlot[x==1,{u,0.1,10},{v,0.1,10},
  PlotStyle->RGBColor[0,0,1],
  PlotPoints->100,DisplayFunction->Identity];
```

```
g6=ImplicitPlot[x==3,{u,0.1,25},{v,0.1,10},
  PlotStyle->RGBColor[0,0.3,0.4],
  PlotPoints->100,DisplayFunction->Identity];
```

```
Show[g2,g3,g4,g5,g6,DisplayFunction->$DisplayFunction];
```



We can describe diffeomorphisms as reversible transformations, which do not generate or smooth out any edges. Now that we have diffeomorphisms as tools, we once more want to take up the already mentioned concept of a “manifold”.

A subset M in $\mathbb{R}^n$ is called k -dimensional submanifold („ k -surface“) in $\mathbb{R}^n$, if for each point p of M there exists an open set U containing p , and a diffeomorphism $f:U \rightarrow V$, where V is an open subset of $\mathbb{R}^n$, such that $f(M \cap U) = \{(y_1, \dots, y_k, 0, \dots, 0)\}$

It is possible to generalize this definition of a manifold as subset of $\mathbb{R}^n$ without needing the surrounding space (s. references, for example: Bröcker or Chillingworth).

At the end of this chapter, we draw a 2-dimensional submanifold of 3-dimensional space: The 2-sphere of radius 1 with the Mathematica statement “ContourPlot3D” (though it is slower than “ParametricPlot3D”. So please be patient while the graphic is calculated and drawn). “Contours” has value 1, if the radius is 1.

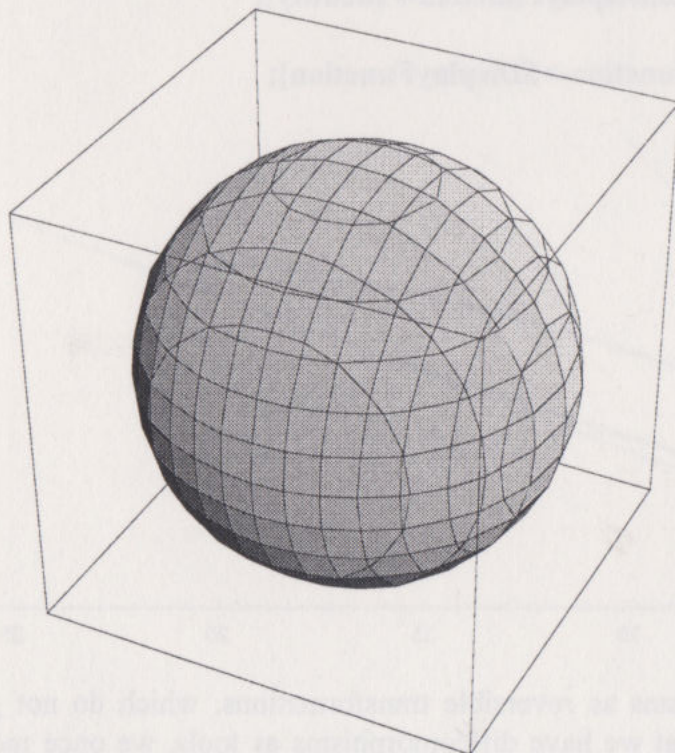
You may use the “Options[ContourPlot3D]” statement, to get the different possibilities for this kind of graphics. (Please do this only after you have used the statement “Needs[“Graphics`ContourPlot3D`”]” as shown in the next lines.) Make experiments with the values of the parameters.

```
Remove["Global`*"]
```

```
Needs["Graphics`ContourPlot3D`"]
```

```
f[x_,y_,z_]:=x^2+y^2+z^2
```

```
ContourPlot3D [f[x,y,z],  
  {x,-7,7}, {y,-4,4}, {z,-5,5},Contours->{1},PlotPoints->{7,10},  
  ViewPoint->{4,2,3}];
```



2 Germs of Functions

The concept of a germ of a map or of a function is used to formulate facts about maps or functions, which are valid only locally, i.e. in some neighborhood of a fixed point.

Two continuous maps $f:U\rightarrow\mathbb{R}^k$ and $g:V\rightarrow\mathbb{R}^k$, defined on neighborhoods U res. V of $p\in\mathbb{R}^n$, are called **equivalent as germs** in p , if there exists a neighborhood $W\subset U\cap V$ on which both coincide.

Maps res. functions, that are equivalent as germs, can be considered to be equal, when we are looking at local features. We examine two examples. In the first example two functions f and

g in two variables x and y are given as follows: $f(x,y)=x$ and $g(x,y)=\begin{cases} x & \text{for } y\leq 1 \\ x+e^{-\frac{1}{y-1}} & \text{for } y>1 \end{cases}$.

In Mathematica the condition for g is set by the operator “/;”.

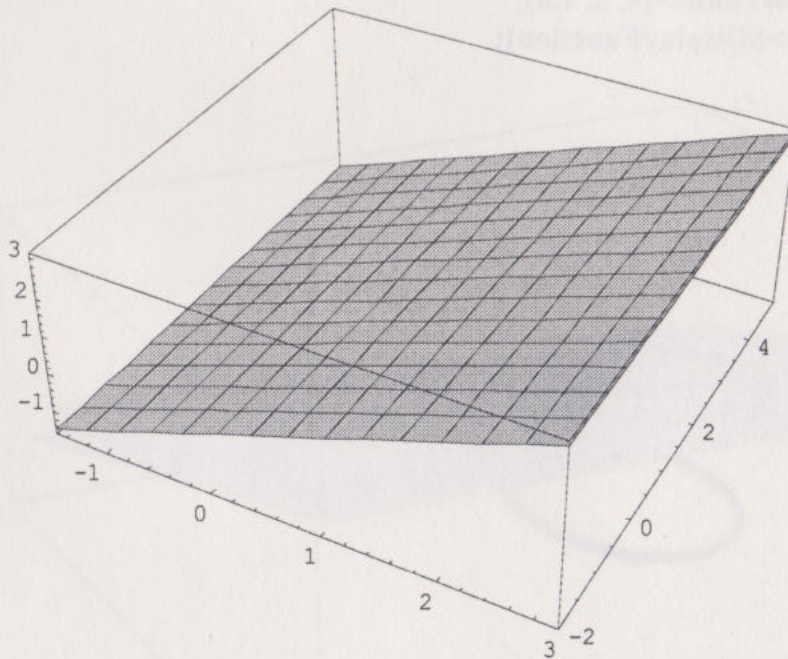
```
f[x_,y_]:=x
```

```
g[x_,y_]:=x/;y<=1
```

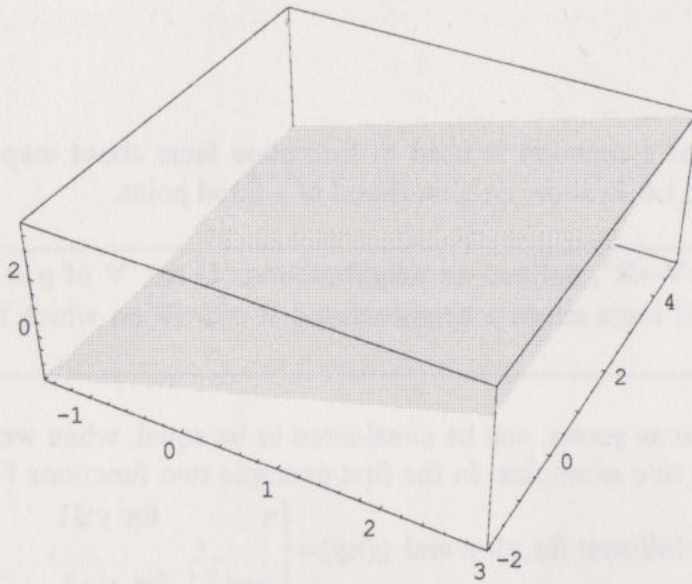
```
g[x_,y_]:=x+Exp[-1/(y-1)]/;y>1
```

First we plot f then g :

```
pic1=Plot3D[f[x,y],{x,-1.5,3},{y,-2,5}];
```



```
pic2=Plot3D[g[x,y],{x,-1.5,3},{y,-2,5},Mesh->False, ColorOutput->CMYKColor];
```



We generate a graphic, which shows an area (circle) on which both functions coincide.

```
u=Cos[t];
```

```
v=Sin[t];
```

```
circ=ParametricPlot3D[{u,v,-3},{Thickness[0.01],  

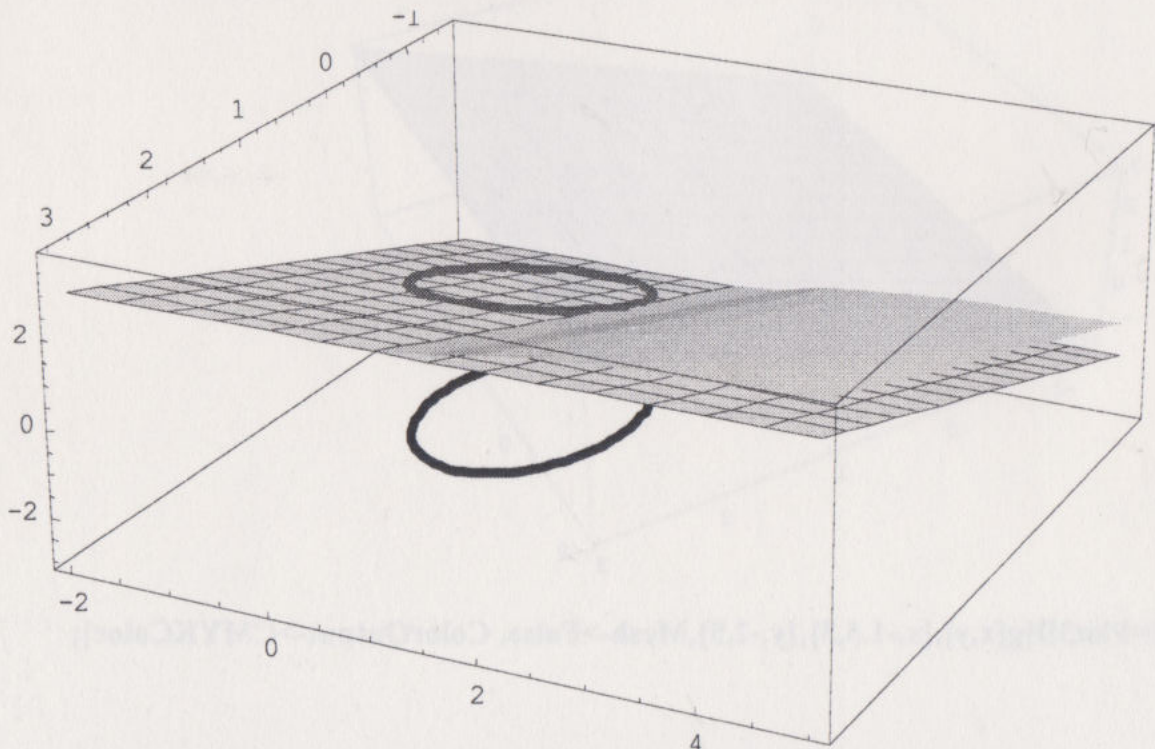
RGBColor[0, 0, 1]},{t,0,2*Pi}, DisplayFunction->Identity];
```

```
circo= ParametricPlot3D[{u,v,u},{Thickness[0.01],  

RGBColor[1, 0,0]},{t,0,2*Pi}, DisplayFunction->Identity];
```

```
Show[pic1,pic2,circ,circo,ViewPoint->{4, 2, 1.8},  

DisplayFunction->$DisplayFunction];
```



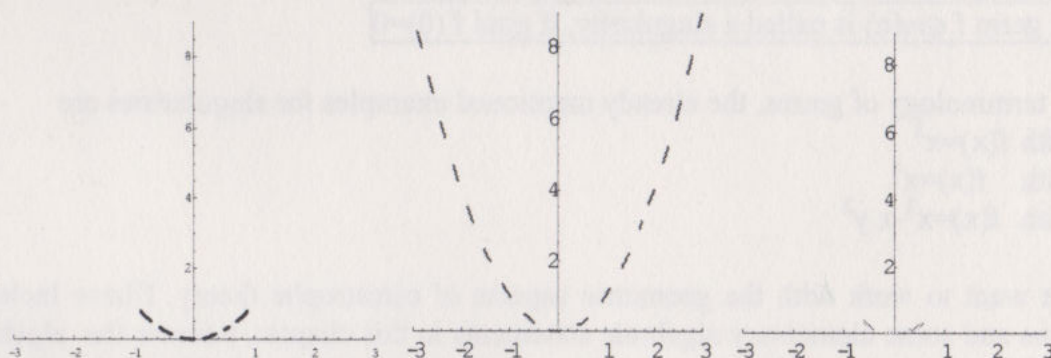
Both functions agree in some neighborhood of the origin (for example the area that appears blue on the display). The red circle only shows the coinciding graphs).

For the second example we choose 3 functions, which have the same formula but different domains:

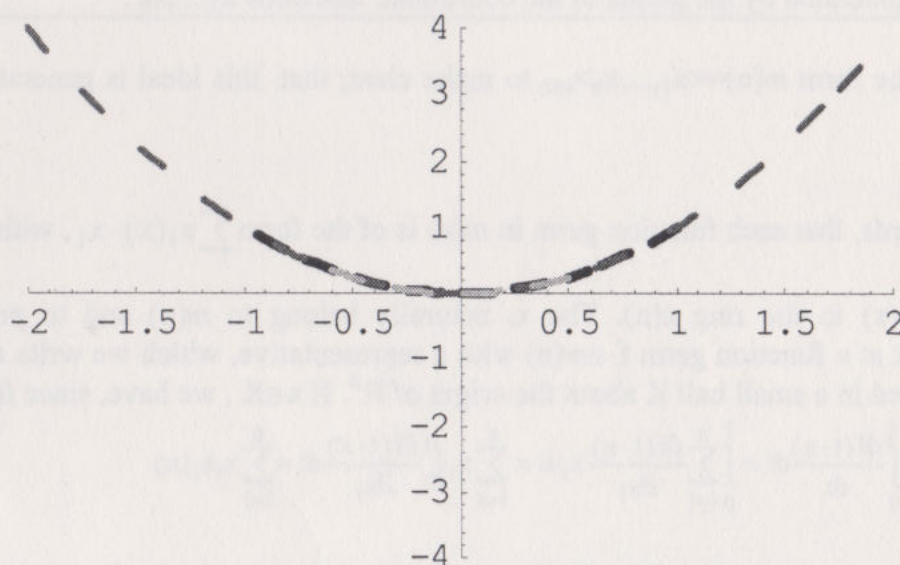
```
g1=Plot[x^2,{x,-1,1},
  PlotStyle->{Dashing[{0.035,0.027}], RGBColor[0,0,1],
  Thickness[0.012]}, PlotRange->{{-3,3},{0,9}}];
```

```
g2=Plot[x^2,{x,-3,3},
  PlotStyle->{Dashing[{0.035,0.061}],RGBColor[1,0,0],
  Thickness[0.01]}, PlotRange->{{-3,3},{0,9}}];
```

```
g3=Plot[x^2,{x,-0.6,0.6},
  PlotStyle->{Dashing[{0.04,0.04}],RGBColor[0,1,0],
  Thickness[0.01]}, PlotRange->{{-3,3},{0,9}}];
```



```
Show[g1,g2,g3,PlotRange->{{-2,2},{-4,4}}]
```



Equivalence as germs is an equivalence relation and the equivalence classes are called map germs or simply germs. We do not make a difference in the notation between maps or functions and their germs. In most books a germ is written in the form $[f]$ or $\tilde{f}$, if f is its

representative.

A map germ is called “smooth” or “ C^∞ ”, if the partial derivatives of any order exist. In what follows, we will understand „differentiable“ always as C^∞ . Often the following notations are found:

The set of all germs of differentiable maps $\mathbb{R}^n \rightarrow \mathbb{R}^k$ at a point p is named $\varepsilon_p(n,k)$. If p is the origin of $\mathbb{R}^n$, one simply writes $\varepsilon(n,k)$ instead of $\varepsilon_0(n,k)$.

Further, if $k=1$, we write $\varepsilon(n)$ instead of $\varepsilon(n,1)$ and speak of function germs (also simplified as “germs”) at the origin of $\mathbb{R}^n$.

For us, the following properties of $\varepsilon(n,k)$ are of interest: $\varepsilon(n,k)$ is a vector space and $\varepsilon(n)$ is an algebra, i.e. a vector space with a structure of a ring. The ring $\varepsilon(n)$ contains a unique maximal ideal, that we call $\mathfrak{m}(n)$. Remember: A subset I of a ring R is called an ideal, if for all elements $a \in I$ and $r \in R$ we have $a \cdot r \in I$. Maximal means, that every other ideal in $R = \varepsilon(n)$ is contained in $\mathfrak{m}(n)$ or equals R . We have $\mathfrak{m}(n) = \{f \in \varepsilon(n) \mid f(0)=0\}$.

We reformulate the notion of a singularity, which we already introduced, for function germs:

A function germ $f \in \mathfrak{m}(n)$ is called a singularity, if $\text{grad } f(0)=0$

In the new terminology of germs, the already mentioned examples for singularities are

$$f \in \mathfrak{m}(1) \text{ with } f(x)=x^2$$

$$f \in \mathfrak{m}(1) \text{ with } f(x)=x^4$$

$$f \in \mathfrak{m}(2) \text{ with } f(x)=x^3 - x y^2$$

Though we want to work with the geometric aspects of catastrophe theory, I have included these notions and some elementary algebraic statements in this chapter, because the algebraic treatment of many problems of catastrophe theory was strongly influenced by the works of J. Mather and can be found everywhere in literature. As an example let's look at the following statement:

The ideal $\mathfrak{m}(n)$ is generated by the germs of the coordinate functions $x_1, \dots, x_n$.

We write this in the form $\mathfrak{m}(n) = \langle x_1, \dots, x_n \rangle_{\varepsilon(n)}$ to make clear, that this ideal is generated over the ring $\varepsilon(n)$.

It says in other words, that each function germ in $\mathfrak{m}(n)$ is of the form $\sum a_i(x) \cdot x_i$, with certain

function germs $a_i(x)$ in the ring $\varepsilon(n)$. The x_i naturally belong to $\mathfrak{m}(n)$ and to proof the statement, we look at a function germ $f \in \mathfrak{m}(n)$ with a representative, which we write as f too, and which is defined in a small ball K about the origin of $\mathbb{R}^n$. If $x \in K$, we have, since $f(0)=0$:

$$f(x) = f(x) - f(0) = \int_0^1 \frac{df(t \cdot x)}{dt} dt = \int_0^1 \sum_{i=1}^n \frac{\partial f(t \cdot x)}{\partial x_i} x_i dt = \sum_{i=1}^n x_i \int_0^1 \frac{\partial f(t \cdot x)}{\partial x_i} dt = \sum_{i=1}^n x_i a_i(x)$$

We will still see further properties of the ideal $\mathfrak{m}(n)$, but first we must learn about k -jets in the next chapter.

3 Jets

The notion of a k-jet of a function germ is a generalization of the Taylor polynomial of order k for functions. Today these notions often are used synonymous.

For a function germ $f \in \varepsilon(n)$ the k -th Taylor polynomial of f at the origin is called the k -jet of f :

$$j^k_0 f(x) = \sum_{|\alpha| \leq k} \frac{1}{\alpha!} \left(\frac{\partial}{\partial x} \right)^\alpha f|_{x=0} \cdot x^\alpha$$

Here α is a multiindex, for which $|\alpha| = \alpha_1 + \dots + \alpha_n$; $\alpha! = \alpha_1! \cdot \dots \cdot \alpha_n!$ and

$$\left(\frac{\partial}{\partial x} \right)^\alpha = \left(\frac{\partial}{\partial x_1} \right)^{\alpha_1} \dots \left(\frac{\partial}{\partial x_n} \right)^{\alpha_n}$$

If, instead of the origin, the point x_0 is the point of development, we write $j^k_{x_0} f$.

We want to realize a jet with Mathematica. For example, let $n=3$ be the order of this polynomial. For a differentiable function f in one variable, the "Series" statement gives the Taylor series:

T=Series[f[x],{x,x0,3}]

$$f[x_0] + f'[x_0] (x - x_0) + \frac{1}{2} f''[x_0] (x - x_0)^2 + \frac{1}{6} f^{(3)}[x_0] (x - x_0)^3 + O[x - x_0]^4$$

The rest is given in the form $O[x-x_0]^4$. We want to adapt the notation to that commonly used in literature: $j^k_{x_0} f$ for the k -jet of f at x_0 . In Mathematica we define a function $j(k, x_0, f, x)$ for the k -jet, where we use the option "Normal" to cancel the series rest:

Remove["Global`*"]

j[k_][x0_][f_][x_] := Series[f[x], {x, x0, k}]/Normal

We compute the 5-jet of a function f at the point 0:

j[5][0][f][x]

$$f[0] + x f'[0] + \frac{1}{2} x^2 f''[0] + \frac{1}{6} x^3 f^{(3)}[0] + \frac{1}{24} x^4 f^{(4)}[0] + \frac{1}{120} x^5 f^{(5)}[0]$$

The 3-jet at the point x_0 is:

j[3][x0][f][x]

$$f[x_0] + (x - x_0) f'[x_0] + \frac{1}{2} (x - x_0)^2 f''[x_0] + \frac{1}{6} (x - x_0)^3 f^{(3)}[x_0]$$

We choose as an example the function $f(x) = (1+x) \sin(x^2)$ and compute the 3-jet at the origin:

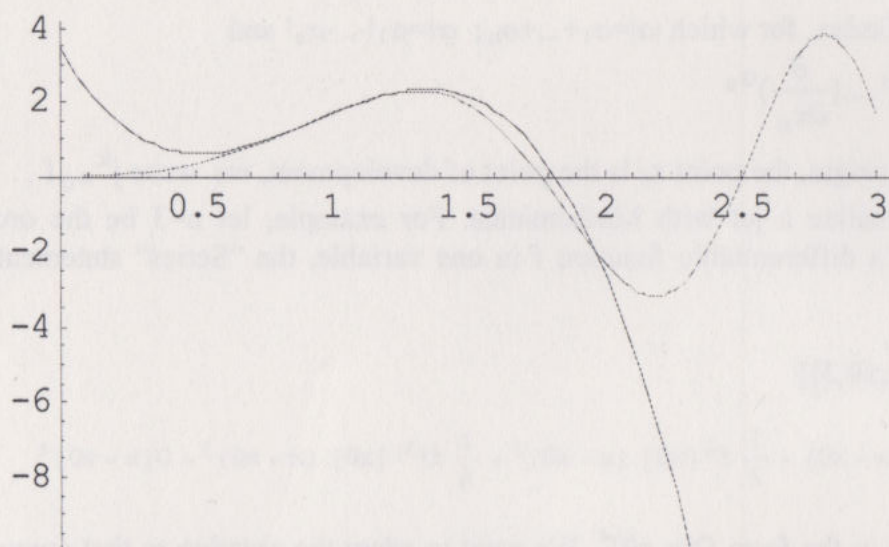
```
f[x_]:= (1+x) Sin[x^2]
```

```
j[3][0][f][x]
```

$$x^2 + x^3$$

We compare the function with its 3-Jet in 1:

```
Plot[{f[x],j[3][1][f][x]//Evaluate,{x,0,3},
PlotStyle->{RGBColor[1,0,0],RGBColor[0,0,1]}};
```



We can see how the jet represents the local behavior of the function.

Functions in two variables can be developed into a Taylor series too. For example the function $f(x,y)=\sin(x \cdot y)$:

```
f[x_,y_]:=Sin[x*y]
```

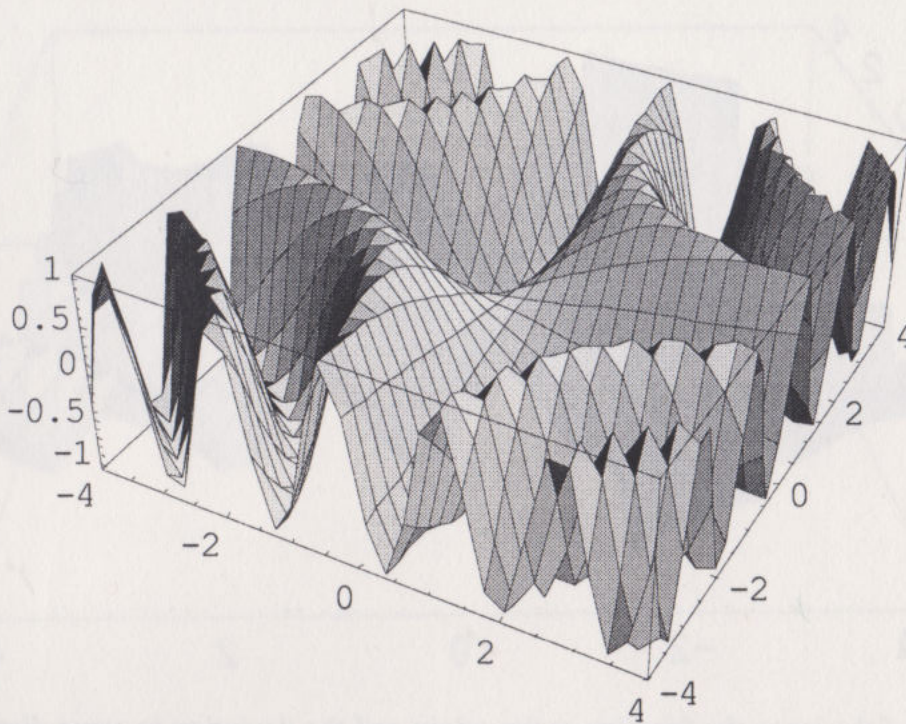
```
j[k1_,k2_][x0_,y0_][f_][x_,y_]:=Series[f[x,y],{x,x0,k1},{y,y0,k2}]
//Normal
```

```
j[7,10][0,0][f][x,y]
```

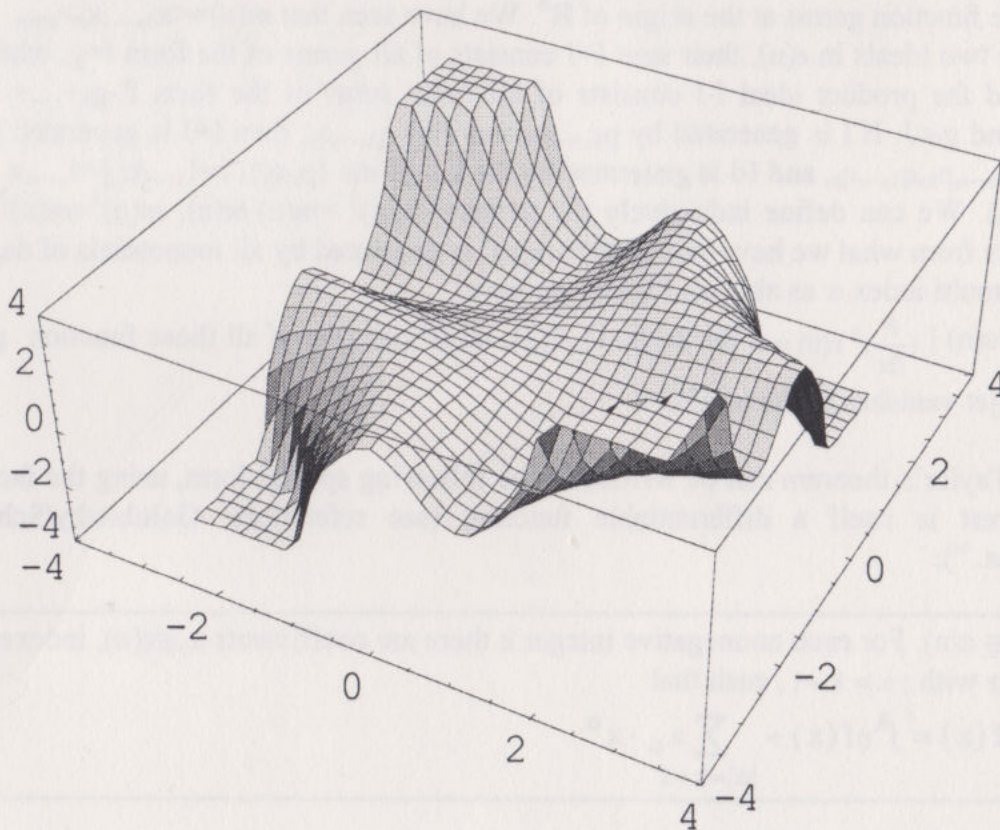
$$x y - \frac{x^3 y^3}{6} + \frac{x^5 y^5}{120} - \frac{x^7 y^7}{5040}$$

Here too we compare both graphs:

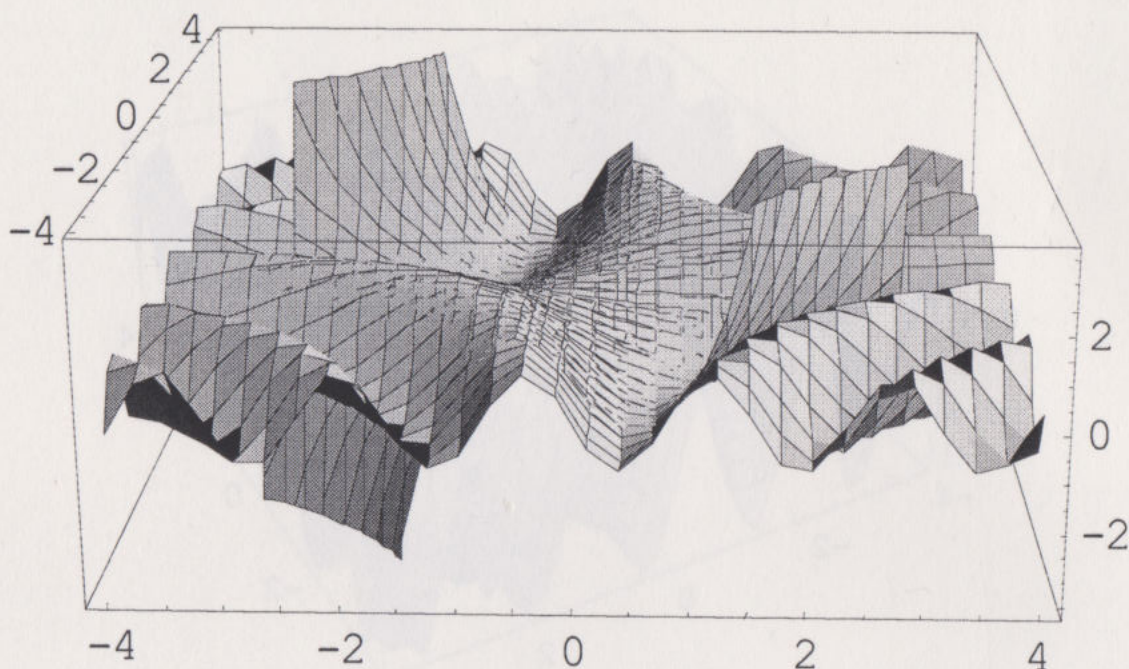
```
b1=Plot3D[f[x,y],{x,-4,4},{y,-4,4},PlotPoints->30]
```

```
b2=Plot3D[Evaluate[j[7,10][0,0][f][x,y]],{x,-3,3},{y,-3,3}, PlotPoints->40,  
PlotRange->{{-4,4},{-4,4},{-4,4}}, ColorOutput->CMYKColor];
```



```
Show[b1,b2,ViewPoint->{0.101, -3.218,1}];
```

The adaptation of the jet to the function at the origin and the deviation in some distance is to be seen.

Now we want to make some additions to the notions about germs, which we introduced in the last chapter. We had $\mathfrak{m}(n) = \{f \in \varepsilon(n) \mid f(0) = 0\}$ as the maximal ideal in the ring of differentiable function germs at the origin of $\mathbb{R}^n$. We have seen that $\mathfrak{m}(n) = \langle x_1, \dots, x_n \rangle_{\varepsilon(n)}$.

If I and J are two ideals in $\varepsilon(n)$, their sum $I+J$ consists of all germs of the form $f+g$, where $f \in I$ and $g \in J$, and the product ideal $I \cdot J$ consists of all finite sums of the form $f_1 \cdot g_1 + \dots + f_m \cdot g_m$, where $f_i \in I$ and $g_i \in J$. If I is generated by $p_1, \dots, p_k$ and J by $q_1, \dots, q_s$, then $I+J$ is generated by the $k+s$ germs $p_1, \dots, p_k, q_1, \dots, q_s$, and $I \cdot J$ is generated by the $k \cdot s$ germs $\{p_i \cdot q_j\}; i=1, \dots, k; j=1, \dots, s$.

Back to $\mathfrak{m}(n)$. We can define inductively the powers: $\mathfrak{m}(n)^2 = \mathfrak{m}(n) \cdot \mathfrak{m}(n)$; $\mathfrak{m}(n)^3 = \mathfrak{m}(n)^2 \cdot \mathfrak{m}(n)$, etc. It follows from what we have said above: $\mathfrak{m}(n)^k$ is generated by all monomials of degree k $\{x^\alpha, |\alpha|=k\}$ (multi index α as above). Finally we have

$\mathfrak{m}(n)^k = \{f \in \varepsilon(n) \mid (\frac{\partial}{\partial x})^\alpha f(0) = 0 \text{ for } |\alpha| \leq k-1\}$, it is, $\mathfrak{m}(n)^k$ consists of all those function germs, whose $(k-1)$ -jet vanishes in the origin.

Remark: Taylor's theorem can be written in the following special form, using the fact, that the series rest is itself a differentiable function (see references: Golubitsky/Schaeffer „Singularities.“):

Taylor: Be $f \in \varepsilon(n)$. For each nonnegative integer k there are coefficients $a_\alpha \in \varepsilon(n)$, indexed by a multi index α with $|\alpha| = k+1$, such that

$$f(x) = j_0^k f(x) + \sum_{|\alpha|=k+1} a_\alpha \cdot x^\alpha$$

4 r-Equivalence

We want to regard function germs as equivalent, if they arise from each other by transformations of their domain, which respect the structure of their singularities. These transformations are diffeomorphisms. We say:

Two function germs $f, g \in \varepsilon(n)$ are called right-equivalent (r-equivalent, written $f \sim g$), if there exists a germ of a local diffeomorphism $h: (\mathbb{R}^n, 0) \rightarrow (\mathbb{R}^n, 0)$, such that

$$f = g \circ h$$

Remark: If, more generally, one admits also a diffeomorphism λ in the range, i.e. $g = \lambda \circ f \circ h$, one speaks of right-left-equivalence. We will restrict ourselves to right-equivalence.

Here an example: Let g be the following function: $g(x) = x^4 - 5x^2 - 3x$

$g[x_] := x^4 - 5x^2 - 3x$

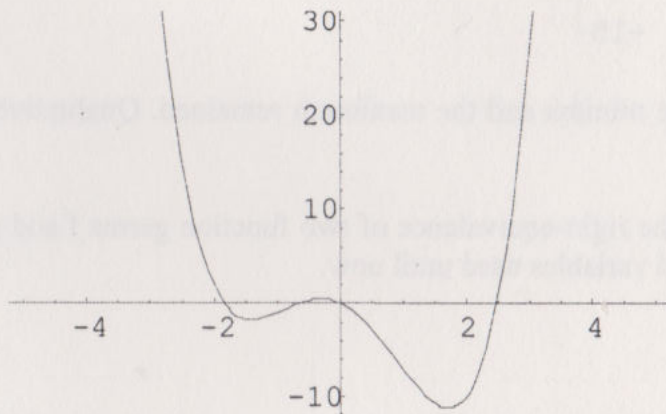
Let h be the diffeomorphism:

$h[x_] := 3x + \text{Exp}[x]$

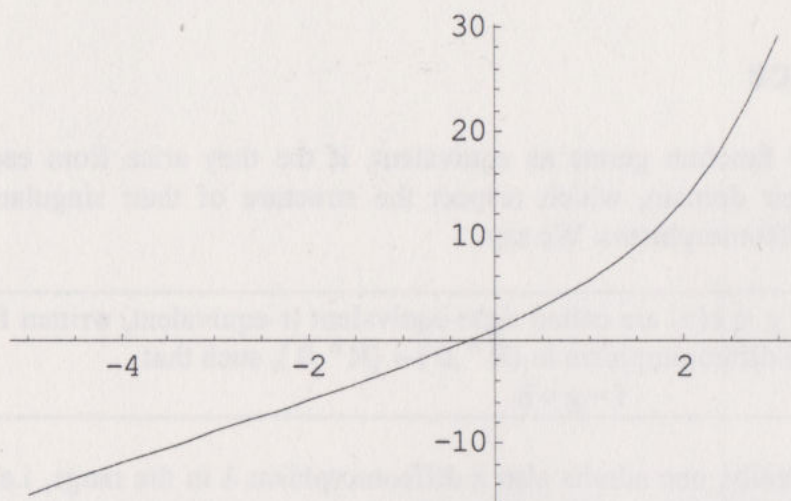
The composition is called f :

$f[x_] := g[h[x]]$

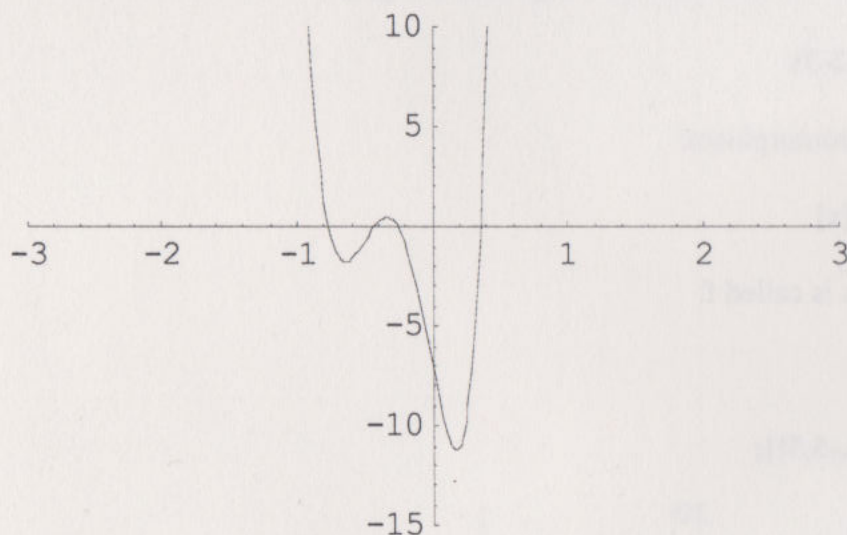
$B1 = \text{Plot}[g[x], \{x, -5, 5\}];$



$B2 = \text{Plot}[h[x], \{x, -5, 3\}];$



```
B3=Plot[f[x],{x,-3,3},PlotRange->{{-3,3},{-15,10}}];
```



As you can see, the configuration of the minima and the maximum remained. Qualitatively f and g are equal.

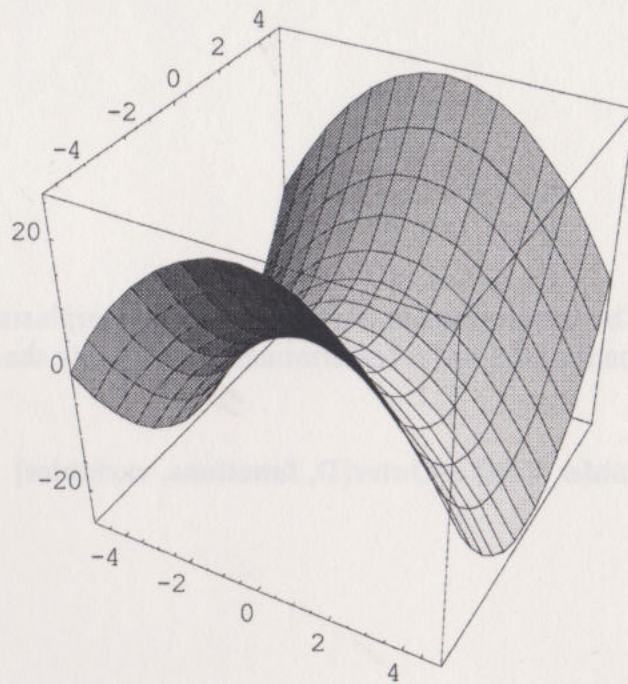
In the next example we want to show the right-equivalence of two function germs f and g in two variables. We clear all functions and variables used until now.

```
Remove["Global`*"]
```

```
Needs["Graphics`Master`"]
```

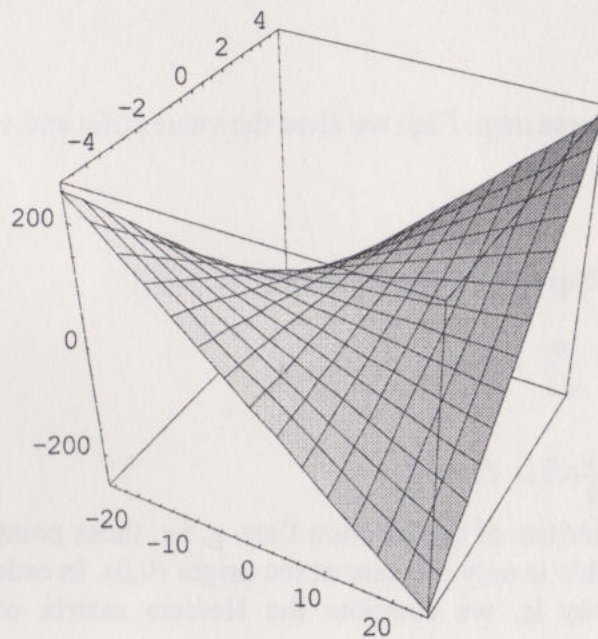
```
f[x_,y_]:= -x^2+y^2
```

```
Plot3D[f[x,y],{x,-5,5},{y,-5,5},BoxRatios->{1,1,1}];
```

$g[x,y]:=2\ x\ y$

Plot3D[g[x,y],{x,-25,25},{y,-5,5},BoxRatios->{1,1,1}, DefaultFont->12];



We look at the following transformation $h:\mathbb{R}^2\rightarrow\mathbb{R}^2$ with $h(x,y)=(u,v)$:

$u=(x+y)/\text{Sqrt}[2];$

$v=(-x+y)/\text{Sqrt}[2];$

Computing $g\circ h=g(h(x,y))=g(u,v)$, one receives

$g[u,v]$

$$(-x + y) (x + y)$$

%//Expand

$$-x^2 + y^2$$

This is just the function f . The transformation $h=(u,v)$ is a diffeomorphism, as the following calculation of the Jacobian matrix of h and its determinant shows, since the determinant is not zero for all (x,y) :

Jacobi[functions_List, variables_List] := Outer[D, functions, variables]

Jh=Jacobi[{u,v},{x,y}];

Jh//MatrixForm

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

Det[Jh]

1

We want to calculate the inverse map. First we clear the values of u and v :

Remove[u,v]

Invmap=Solve[{u==(x+ y)/Sqrt[2],v==(-x+ y)/Sqrt[2]},{x,y}]

$$\left\{ \left\{ x \rightarrow \frac{1}{2} (\sqrt{2} u - \sqrt{2} v), y \rightarrow \frac{u}{\sqrt{2}} + \frac{v}{\sqrt{2}} \right\} \right\}$$

We have $(x,y) = h^{-1}(u,v) = \left(\frac{1}{2}(\sqrt{2} u - \sqrt{2} v), \frac{u}{\sqrt{2}} + \frac{v}{\sqrt{2}} \right)$

Now we calculate the singularities of the function f res. g , i.e. those points, where the gradient vanishes. We will see, that this is only the case at the origin $(0,0)$. In order to make clear what kind of point the singularity is, we compute the Hessian matrix of f res. g and their eigenvalues.

gradf=Jacobi[{f[x,y]},{x,y}]

$$\{ \{-2 x, 2 y\} \}$$

gradg=Jacobi[{g[x,y]},{x,y}]

$$\{ \{2 y, 2 x\} \}$$

Indeed the gradient vanishes only for $x=y=0$.


```
{f1[x_,y_],f2[x_,y_]}=Flatten[gradf];
```

```
{g1[x_,y_],g2[x_,y_]}=Flatten[gradg];
```

```
Hf[x_,y_]:=Evaluate[{{D[f1[x,y],x],D[f1[x,y],y]},  
                  {D[f2[x,y],x],D[f2[x,y],y]}}
```

```
Hf[x,y]//MatrixForm
```

$$\begin{pmatrix} -2 & 0 \\ 0 & 2 \end{pmatrix}$$

```
Eigenvalues[Hf[x,y]]
```

$$\{-2, 2\}$$

```
Hg[x_,y_]:=Evaluate[{{D[g1[x,y],x],D[g1[x,y],y]},  
                  {D[g2[x,y],x],D[g2[x,y],y]}}
```

```
Hg[x,y]//MatrixForm
```

$$\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}$$

```
Eigenvalues[Hg[x,y]]
```

$$\{-2, 2\}$$

The Hessian matrix of f res. g has a positive and a negative eigenvalue. Thus the singularity is a saddle point.

At the end of this chapter we want to speak about “determinacy“, which plays an important roll in catastrophe theory. In our geometric approach it will be introduced in short.

A function germ $f \in \mathcal{E}(n)$ is called k-determined, if every other function germ with the same k-jet is right-equivalent to f .

Especially, a k-determined function germ is right-equivalent to its own k-jet, i.e. there exists a diffeomorphism, by which the function-germ may be written as a polynomial of degree k . Whether a function germ is k-determined or not, can be proofed by the so called “determinacy test“. This test (J. Mather) exists in different formulations. One of these, a practical “checklist“, can be found in the book of H. Ursprung (see reference).

5 The local behavior of smooth maps

Catastrophe theory contains in its mathematical kernel the elements of singularity theory. A comprehensive introduction to singularity theory is Lu's book (see reference). We want to make a classification of the germs of differentiable maps, as it is done in singularity theory. An enlargement of this division will be the classification theorem of R. Thom, which is quasi the heart of catastrophe theory. We will describe it in an extra chapter.

The first essential criterion for a classification of the local behavior of a map germ $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ in a point x_0 of the domain, is the rank of the Jacobian matrix:

$$J_{x_0} f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_0) & \dots & \frac{\partial f_1}{\partial x_n}(x_0) \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(x_0) & \dots & \frac{\partial f_m}{\partial x_n}(x_0) \end{pmatrix}$$

Recall: A point x_0 is called regular, if $\text{rank } J_{x_0} f = \text{maximal} = \min(m, n)$, else x_0 is called singular.

We classify:

1.) x_0 is a regular point of f , then we have three cases:

a) $n > m$. Then f is local, it is, in a neighborhood U of x_0 , a submersion (a differentiable map $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ with $\text{rank } J_{x_0} f = m$) and " f looks locally like" a projection $f \sim y = (y_1, \dots, y_m)$, (for the notions submersion and immersion see references, Bröcker/Jänich)

b) $n < m$. Then f locally is an immersion (a differentiable map $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ with $\text{rank } J_{x_0} f = n$) and we have: $f \sim y = (y_1, \dots, y_m, 0, \dots, 0)$.

c) $n = m$: f is locally a diffeomorphism, it is, f looks locally like the identical map: $f \sim \text{id}$.

2.) x_0 is singular, it is, $\text{rank } J_{x_0} f < \min(m, n)$, then for general m and n no statements can be made. Only for $m = 1$, that is for functions, we have:

Assume $\text{rank } J_{x_0} f < 1$, thus $\text{rank } J_{x_0} f = 0$, or equivalent to this $\text{grad } f(x_0) = 0$. Then, for a further classification, we must look at the Hessian matrix of the second order partial derivatives:

$$H_{x_0} f = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{\substack{i=1, \dots, n \\ j=1, \dots, m}} \Big|_{x=x_0}$$

Case a) $H_{x_0} f$ is regular, i.e. $\text{rank } H_{x_0} f = n$, or $\det H_{x_0} f \neq 0$. Then f can be transformed into the so called Morse normal form: $f \sim \pm x_1^2 \dots \pm x_n^2$ (Morse lemma). The interested reader can find the exact procedure in the book of Poston and Stewart: "Catastrophe Theory.." (see reference).

Case b) $\text{rank } H_{x_0} f < n$, or $\det H_{x_0} f = 0$, then we have the splitting lemma (Gromoll-Meyer lemma): $f \sim \pm x_1^2 \dots \pm x_k^2 + N$, where k is the rank of the Hessian matrix and N is the so called non-Morse part, a degenerated (i.e. its Hessian determinant vanishes) term of higher order. Concerning this term, elementary catastrophe theory makes more statements, if its codimension (see below) is at most 4.

Note the essential statement of the splitting lemma under the following viewpoint: If a function germ, depending on several variables, has a degenerated critical point, then the rank of the Hessian matrix in that point decides, in how many space dimensions the germ can be regarded as non-degenerated, because such a number of quadratic terms can be splitted. Only as many space dimensions as

$$\text{corank} = n - \text{rank } H_{x_0} f$$

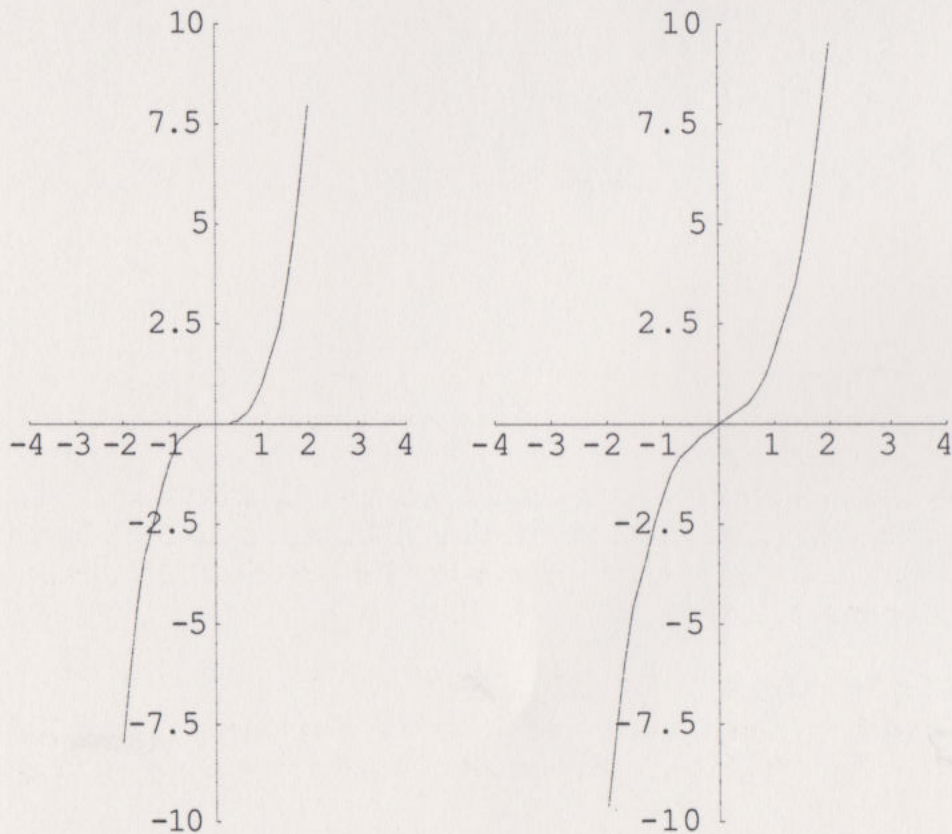
gives, are concerned with the “degeneracy”. Only these space variables contain the “structural instability” (see also next chapter) of the germ.

6 Unfoldings

Look at the singularity $f(x)=x^3$. We add to it a linear “disturbance” $u \cdot x$, where the factor (parameter) u may assume different values. For a first overview, we draw the disturbed function $x^3 + u \cdot x$ for example for values of the parameters from 0 to 4 in steps of 0.2. From the graphics generated by Mathematica, we only show two.

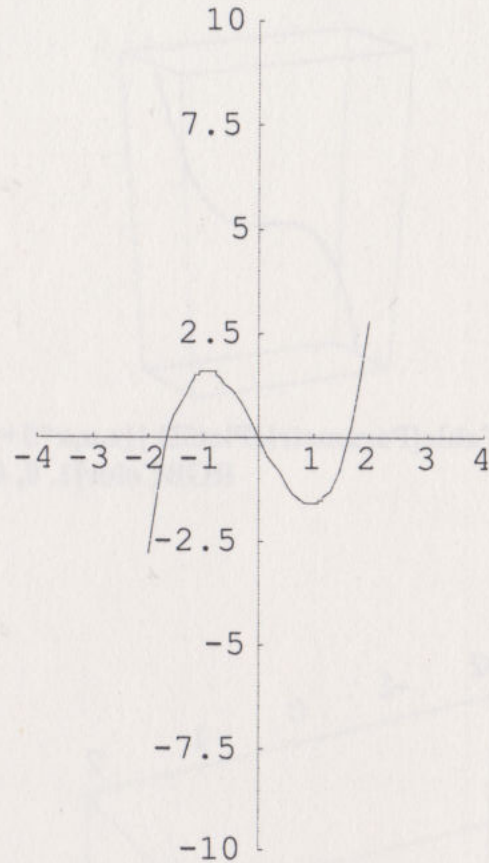
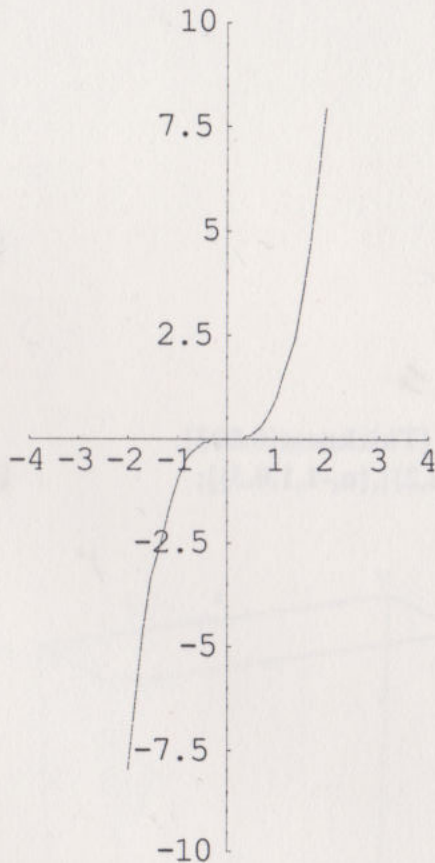
```
Remove["Global`*"]
```

```
Do[Plot[x^3+u x,{x,-2,2},AspectRatio->Automatic,  
PlotRange->{{-4,4},{-10,10}},{u,0,4,0.2}]
```



We do the same for parameter values from 0 to -4 in steps of -0.2:

```
Do [Plot[x^3+u x,{x,-2,2},AspectRatio->Automatic,  
PlotRange->{{-4,4},{-10,10}},{u,0,-4,-0.2}]
```

Please note: While for positive parameter values the disturbed function has no singularity, in case of negative u -values a relative maximum and a relative minimum exists. The origin is only for $u=0$ a singular point.

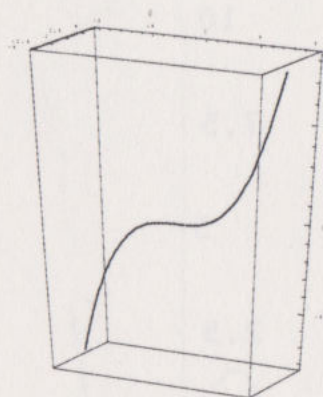
We now show, how one can think of f as a member of a function family. To do so, we first draw the graph of the function $x^3+u \cdot x$ with the "Plot3D" statement. Then we draw the disturbed functions as single space curves with "ParametricPlot3D". Finally we show all objects together in one graphic.

```
pic=Plot3D[x^3+u x,{x,-2,2},{u,-2,1.5},AspectRatio->2,
BoxRatios->{1,1,2},Boxed->True,PlotPoints->30,ViewPoint->{0,-2.4,2.},Mesh->False,
AxesLabel->{"x","u","F"},DefaultFont->10,DisplayFunction->Identity];
```

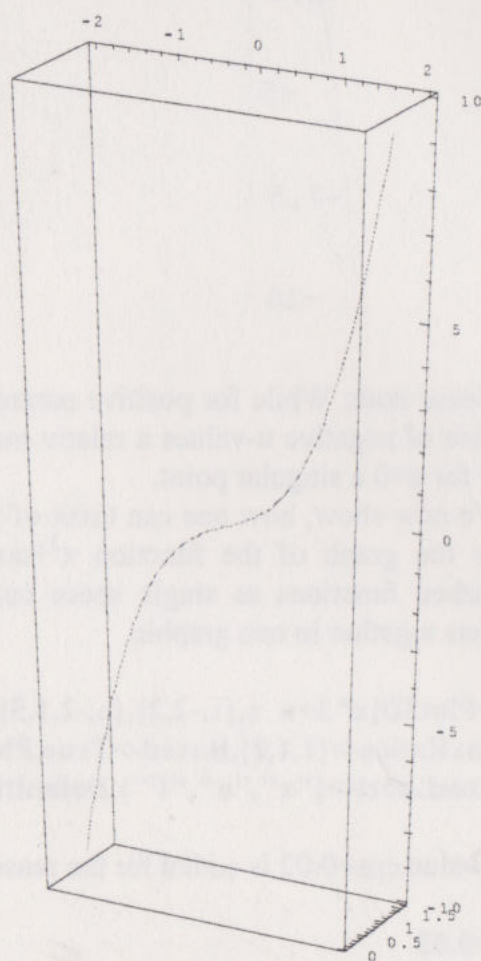
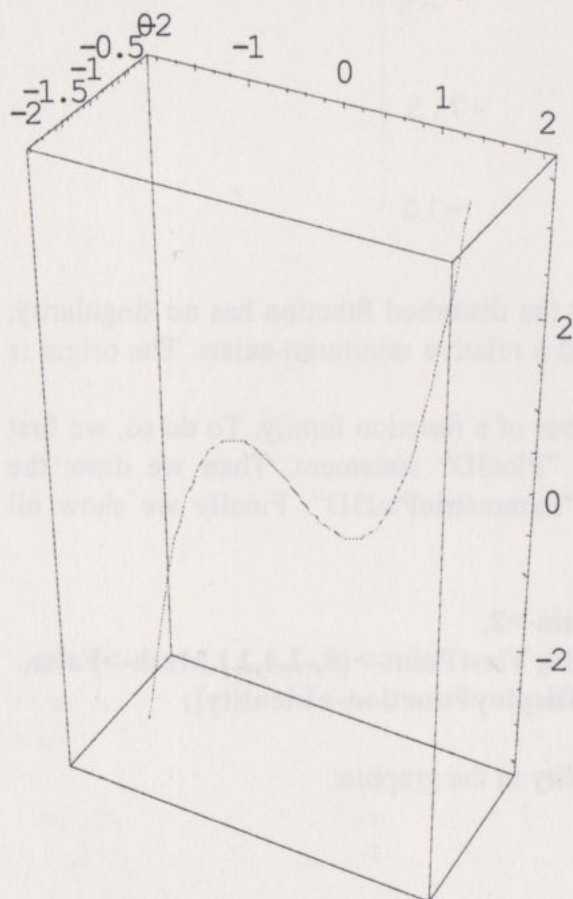
The value $\text{eps}=0.02$ is added for the reason of visibility in the graphic:

```
eps=0.02
```

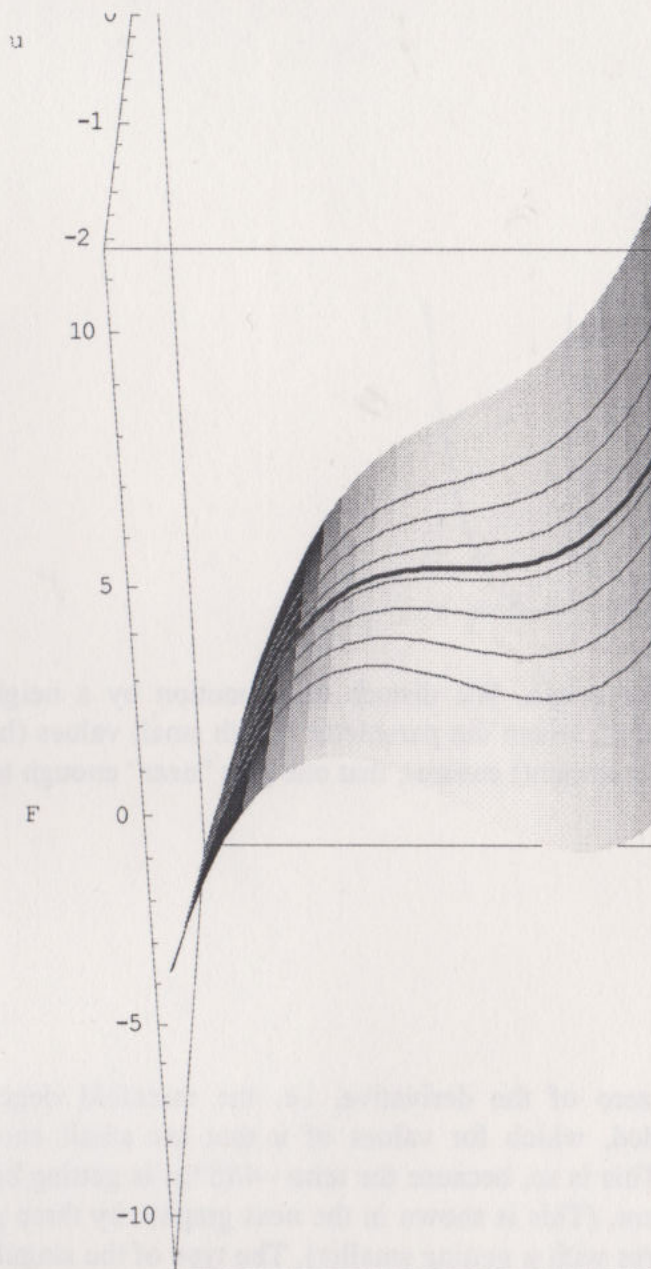
```
center=ParametricPlot3D[{x,0,x^3+eps,
{Thickness[0.008],RGBColor[0,0,1]}},{x,-2,2}]
```



```
sections=Table[ParametricPlot3D[{x,u,x^3+u x+eps,{Thickness[0.003],
RGBColor[1, 0, 0]}},{x,-2,2},{u,-1,1,0.3}];
```



```
Show[pic,sections,center,ColorOutput->CMYKColor,
DefaultFont->12, DisplayFunction->$DisplayFunction]
```

The type of singularity of the function $f(x)=x^3$ changes, as we have seen, with the addition of a small disturbing function. This behavior of f is called “structurally unstable”. But $f(x)=x^3$ can be seen as a member of a whole family of functions $F(x,u)=x^3+u \cdot x$, because $F(x,0)=f(x)$. This family is stable in that sense, that all kinds of singularities of its members are included. This family is only one of the different possibilities to “unfold” the function $f(x)=x^3$ by adding disturbing terms. For example $F(x,u)=x^3+u x^2+v x$ would be another such unfolding of f .

An unfolding of a differentiable germ $f \in \mathcal{M}(n)$ is a germ $F \in \mathcal{M}(n+r)$ with $F|_{\mathbb{R}^n} = f$. r is called unfolding dimension of f .

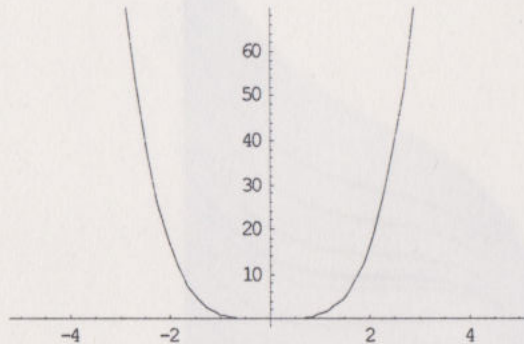
Another example: $f(x)=x^4$ with the unfolding $F(x,u,v)=x^4 + u x^2 + v x$. Here is $n=1$ and $r=2$.

Now it's the question, whether one can find unfoldings, which quasi contain all possible disturbances (all types of singularities), i.e. which are stable and moreover have a minimum

number of parameters.
Look at the singularity x^4 :

$$f[x]=x^4$$

Plot[f[x],{x,-5,5}];



It has a degenerated minimum at the origin. We disturb this function by a neighboring polynomial of higher degree, e.g. by $u \cdot x^5$, where the parameter u with small values (here and in what follows “small” means small in amount) ensures, that one gets “near” enough to x^4 .

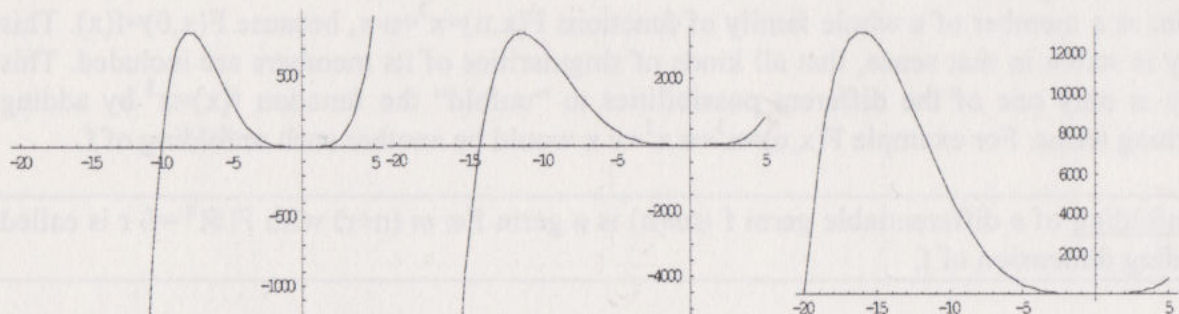
$$f[x]:=x^4+u \cdot x^5$$

Solve[f'[x]==0,x]

$$\left\{ \{x \rightarrow 0\}, \{x \rightarrow 0\}, \{x \rightarrow 0\}, \left\{x \rightarrow -\frac{4}{5u}\right\} \right\}$$

As you can see, to the threefold zero of the derivative, i.e. the threefold degenerated minimum, another extremum is added, which for values of u that are small enough, is arbitrarily far away from the origin. This is so, because the term $-4/(5 \cdot u)$ is getting bigger in amount for u getting smaller in amount. (This is shown in the next graphic by three pictures from the generated sequence of pictures with u getting smaller). The type of the singularity in the origin thus isn't influenced by neighboring polynomials of the form $u \cdot x^5$.

Pic2=Do[Plot[f[x],{x,-20,5}],{u,0.1,0.05,-0.01}]



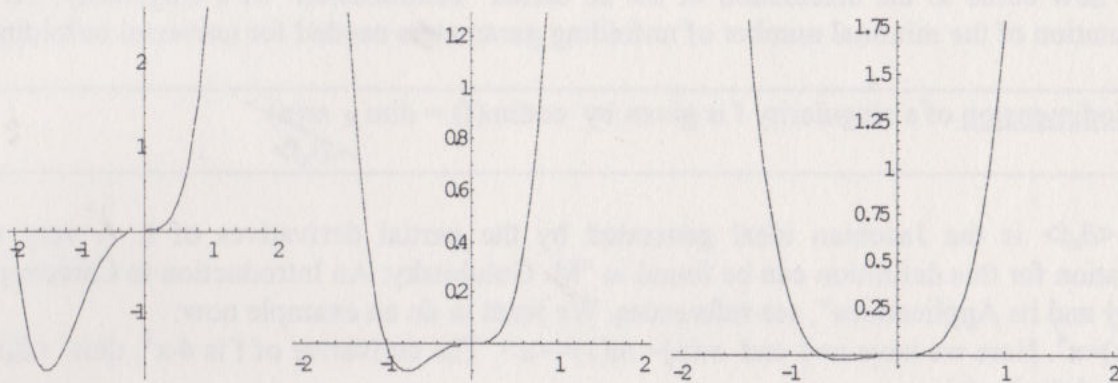
We get quite another picture, if we disturb the polynomial x^4 by a neighboring polynomial of lower degree, e.g. $u \cdot x^3$. Here for small amounts of u a new minimum is generated arbitrarily close to the existing singularity at the origin.

$$g[x_]:=x^4+u x^3$$

$$\text{Solve}[g'[x]==0,x]$$

$$\left\{ \{x \rightarrow 0\}, \{x \rightarrow 0\}, \left\{x \rightarrow -\frac{3u}{4}\right\} \right\}$$

$$\text{Bild2}=\text{Do}[\text{Plot}[g[x],\{x,-2,2\}],\{u,2,0.1,-0.1\}]$$



If disturbed by a linear term, the singularity at the origin can even be eliminated.

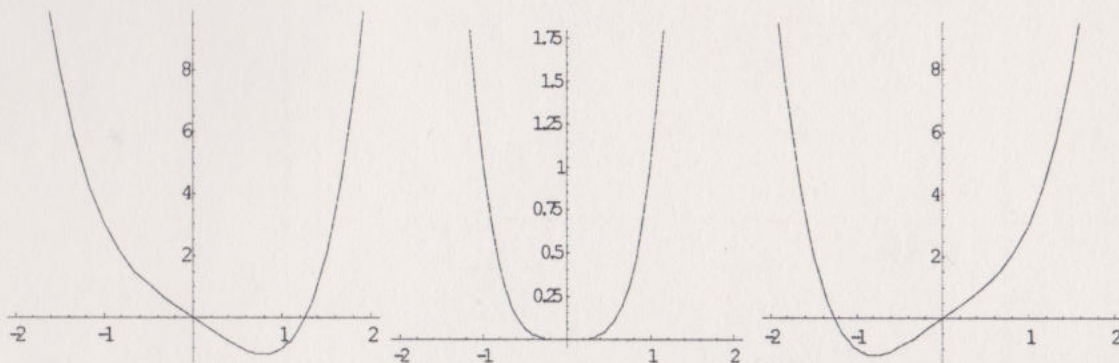
$$h[x_]:=x^4+u x$$

$$\text{Solve}[h'[x]==0,x]$$

$$\left\{ \left\{x \rightarrow -\frac{u^{1/3}}{2^{2/3}}\right\}, \left\{x \rightarrow \frac{(-1)^{1/3} u^{1/3}}{2^{2/3}}\right\}, \left\{x \rightarrow -\frac{(-1)^{2/3} u^{1/3}}{2^{2/3}}\right\} \right\}$$

We note: Only for $u=0$, i.e. for a vanishing disturbance, h is singular at the origin. For each $u \neq 0$ the linear disturbance ensures, that no singularity at the origin occurs.

$$\text{pic2}=\text{Do}[\text{Plot}[h[x],\{x,-2,2\}],\{u,-2,2,0.2\}]$$



The function $f(x)=x^4$ is structurally unstable. To find a stable unfolding for the singularity $f(x)=x^4$, there must be added terms of lower order. But we need not take into account all terms $x^4+u \cdot x^3+v \cdot x^2+w \cdot x+k$, since the absolute term plays no roll for the computation of the singular points, and each polynomial of degree 4 can be written by a suitable change of

coordinates without cubic term (similar as a polynomial of third degree, after a coordinate transformation, can always be written without quadratic term ("Tschirnhaus-transformation")). Indeed, the expression $F(x,u,v)=x^4+u\cdot x^2+v\cdot x$ is, as we shall see later, the "universal" unfolding of x^4 . Here an unfolding F of a germ f is called universal, if every other unfolding of f can be received by suitable coordinate transformations "morphism of unfoldings", see next chapter) from F , and the number of unfolding parameters of F is minimal.

We now come to the calculation of the so called "codimension" of a singularity, i.e. the computation of the minimal number of unfolding parameters needed for universal unfoldings:

The codimension of a singularity f is given by $\text{codim}(f) = \dim_{\mathbb{R}} \frac{m(n)}{\langle \partial_x f \rangle}$

Here $\langle \partial_x f \rangle$ is the Jacobian ideal generated by the partial derivatives of f . A very nice motivation for this definition can be found in "M. Golubitsky: An Introduction to Catastrophe-Theory and its Applications", see references. We want to do an example now:

Be $f(x)=x^4$. Here we have $n=1$ and $m(n)=m(1)=\langle x \rangle$. The derivative of f is $4\cdot x^3$, thus $\langle \partial_x f \rangle = \langle x^3 \rangle$ and we have

$$\text{codim}(f) = \dim_{\mathbb{R}} \frac{m(1)}{\langle \partial_x f \rangle} = \dim_{\mathbb{R}} \frac{\langle x \rangle}{\langle x^3 \rangle} = 2$$

So the universal unfolding of x^4 needs two unfolding parameters. We will see in the next chapter, how to find this unfolding.

7 The seven elementary catastrophes

Elementary catastrophe theory works with families (unfoldings) of potential functions

$$F: \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}, \text{ with } k \leq 4$$

$$(x, u) \mapsto F(x, u)$$

$\mathbb{R}^n$ is called state space, $\mathbb{R}^k$ is called parameter space or control space. Accordingly $x=(x_1, \dots, x_n)$ is called state variable or endogenous variable and $u=(u_1, \dots, u_k)$ is called control variable (exogenous variable, parameter).

We are interested in the singularities of F with respect to x and in dependence of the control variables u . We regard F as unfolding of a function germ and want to classify F .

Let's speak again about the already mentioned notion of a universal unfolding. At first we give a definition that looks at first glance a little bit complicated. This definition is a generalization of diffeomorphisms and r -equivalence between function germs to families of parameterized function germs, which respect the structure (fibers) of $\mathbb{R}^n \times \mathbb{R}^k$.

Suppose $f \in \varepsilon(n)$ and $F: \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}$ and $G: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}$ be unfoldings of f . A right-morphism from F to G , also called unfolding morphism, is a pair (Φ, α) , with $\Phi \in \varepsilon(n+k, n+r)$ and $\alpha \in \mathcal{M}(k)$, such that:

- 1.) $\Phi|_{\mathbb{R}^n} = \text{id}(\mathbb{R}^n)$, that is $\Phi(x, 0) = (x, 0)$
- 2.) If $\Phi = (\phi, \psi)$, with $\phi \in \varepsilon(n+k, n)$, $\psi \in \varepsilon(n+k, r)$, then $\psi \in \varepsilon(k, r)$
- 3.) For all $(x, u) \in \mathbb{R}^n \times \mathbb{R}^k$ we have

$$F(x, u) = G(\Phi(x, u)) + \alpha(u)$$

In analogy to right-equivalence of function germs, we call two unfoldings F and G of the same function germ right-equivalent as unfoldings, if there exists a right-morphism from F to G .

An unfolding F of f is called versal (or stable), if each other unfolding of f is right-equivalent as unfolding to F , i.e., if there exists a right-morphism between the unfoldings

A versal unfolding with minimal unfolding dimension is called universal unfolding of f .

We have the following theorem:

Theorem on the existence of a universal unfolding: A singularity f has a universal unfolding iff $\text{codim}(f) = k < \infty$

The proofs of the theorems in this book can be found in the books of Th.Bröcker or G.Wassermann (see references).

Examples (see also the following theorem with its examples):

If $f(x) = x^3$, then we have

$F(x, u) = x^3 + u x^2 + v x$ is versal,

$F(x, u) = x^3 + u x$ is universal,

$F(x, u) = x^3 + u x^2$ is not versal.

The following theorem says, how one can find an universal unfolding of a function germ:

Theorem on the normal form of universal unfoldings

Be $f \in \mathcal{m}(n)$ a singularity with $\text{codim}(f) = k < \infty$. Be $u = (u_1, \dots, u_k)$ the parameters of the unfolding. Be $b_i(x)$, $i=1, \dots, k$, elements of $\mathcal{m}(n)$, whose cosets modulo $\langle \partial_x f \rangle$ generate the vector space $\mathcal{m}(n) / \langle \partial_x f \rangle$. Then

$$F(x, u) = f(x) + \sum_{i=1}^k u_i b_i(x)$$

is a universal unfolding of f .

Examples :

a) Be $f(x) = x^4$. As we have already seen in the last chapter, $n=1$ and $\mathcal{m}(n) = \mathcal{m}(1) = \langle x \rangle$. The derivative of f is $4 \cdot x^3$, thus $\langle \partial_x f \rangle = \langle x^3 \rangle$ and we have $\langle x \rangle / \langle x^3 \rangle = \langle x, x^2 \rangle$. Thus $F(x, u_1, u_2) = x^4 + u_1 x^2 + u_2 x$ is the universal unfolding of f .

b) Be $f(x, y) = x^3 + y^3$. Here $n=2$ and $\mathcal{m}(n) = \mathcal{m}(2) = \langle x, y \rangle$. The partial derivatives of f by x res. y are $3x^2$ and $3y^2$, thus $\langle \partial_x f \rangle = \langle x^2, y^2 \rangle$ and we have $\langle x, y \rangle / \langle x^2, y^2 \rangle = \langle x, xy, y \rangle$. Thus $F(x, y, u_1, u_2, u_3) = x^3 + y^3 + u_1 \cdot x \cdot y + u_2 \cdot x + u_3 \cdot y$ is the universal unfolding of f .

Hint: For the calculation of the quotient space there is a useful method, which you will find in literature as "Siersma's trick".

We now present René Thom's famous list of the seven elementary catastrophes. This is followed by some graphics, one for each of the centers of organization in the list. The universal unfoldings are described in the following chapters.

Classification theorem (Thom's list)

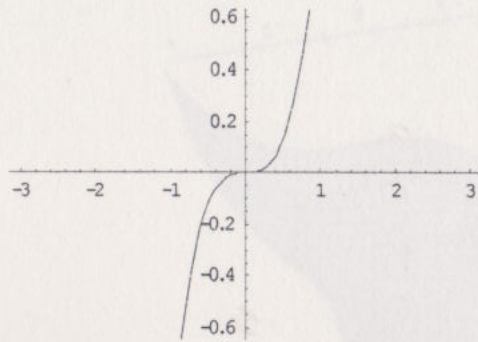
Up to addition of a non degenerated quadratic form in other variables and up to multiplication by ± 1 a singularity f of codimension k ($1 \leq k \leq 4$) is right-equivalent to one of the following seven:

f	$\text{codim } f$	universal unfolding	name
x^3	1	$x^3 + u x$	fold
x^4	2	$x^4 + u x^2 + v x$	cuspid, Riemann-Hugoniot catastrophe
x^5	3	$x^5 + u x^3 + v x^2 + w x$	swallowtail
$x^3 + y^3$	3	$x^3 + y^3 + u x y + v x + w y$	hyperbolic umbilic
$x^3 - x y^2$	3	$x^3 - x y^2 + u(x^2 + y^2) + v x + w y$	elliptic umbilic
x^6	4	$x^6 + u x^4 + v x^3 + w x^2 + t x$	butterfly
$x^2 y + y^4$	4	$x^2 y + y^4 + u x^2 + v y^2 + w x + t y$	parabolic umbilic

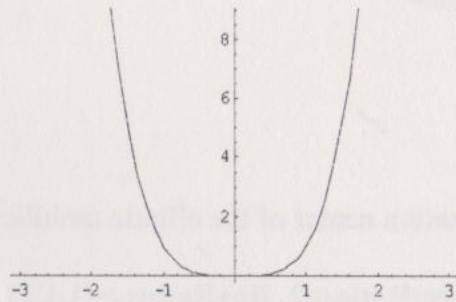
In short: The 7 elementary catastrophes are the 7 universal unfoldings of singular function germs of codimension k ($1 \leq k \leq 4$). The singularities are called organization centers of the catastrophes.

In the next chapter we will look at the geometry of the 7 elementary catastrophes. Now let's show graphically how their organization centers (first column of the list) look like. At first we draw the graphs of the functions in a single variable:

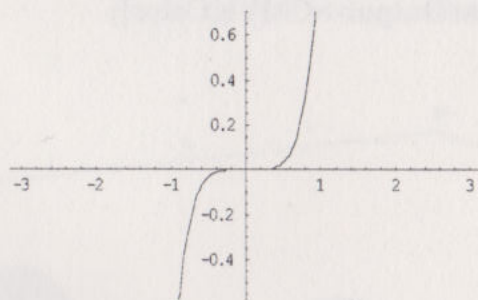
Plot[x^3,{x,-3,3}];



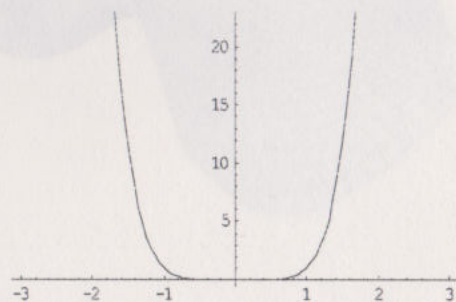
Plot[x^4,{x,-3,3}];



Plot[x^5,{x,-3,3}];

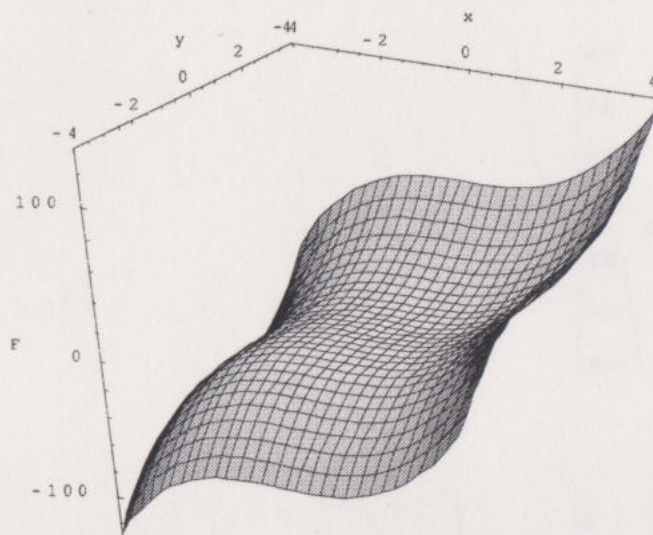


Plot[x^6,{x,-3,3}];



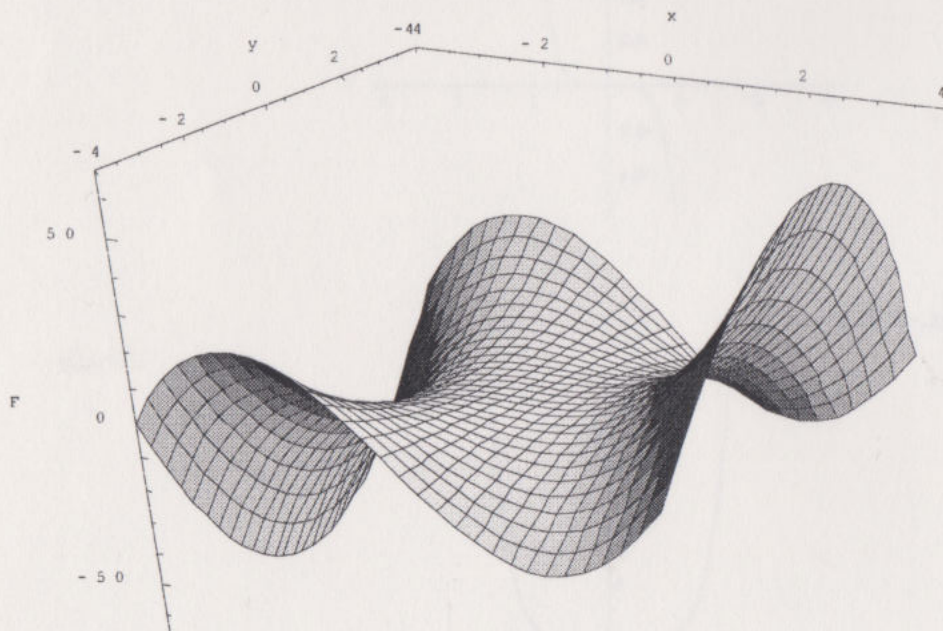
Now we draw the graphs of the functions from the list, which have two independent variables x and y . At first we draw the organization center of the so called hyperbolic umbilic:

**Plot3D[x^3+y^3,{x,-4,4},{y,-4,4}, AspectRatio->2,BoxRatios->{1,1,2},
Boxed->False,PlotPoints->30, ViewPoint->{1.3,-2.4,2.},Mesh->True,
AxesLabel->{"x","y","F"},ColorOutput->CMYKColor];**



The next one is the graphic for the organization center of the elliptic umbilic:

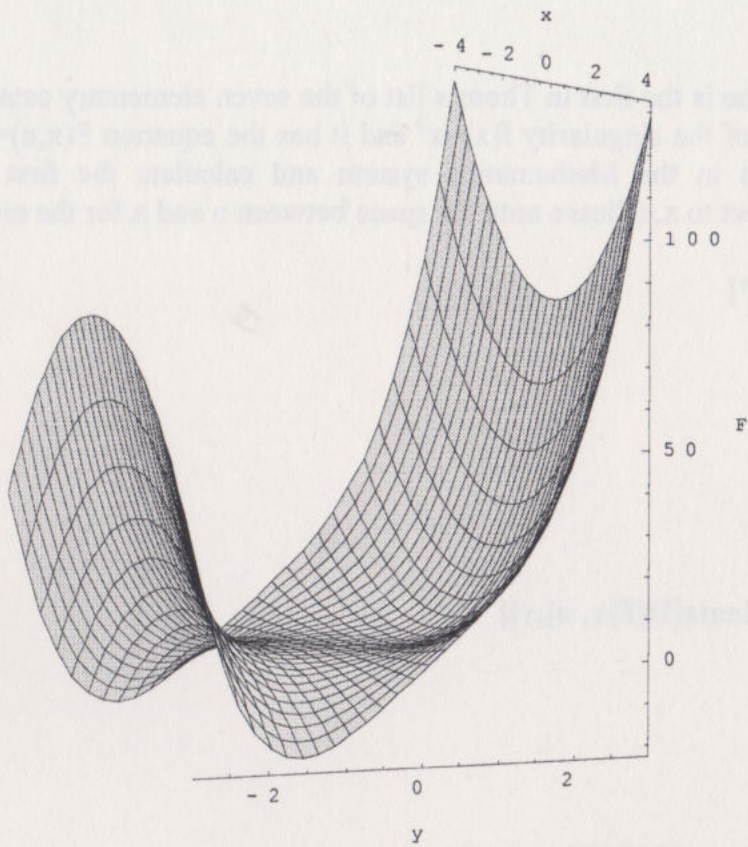
```
Plot3D[x^3-x y^2,{x,-4,4},{y,-4,4}, AspectRatio->2, BoxRatios->{1,1,2},
Boxed->False,PlotPoints->30, ViewPoint->{1.3,-2.4,2.},Mesh->True,
AxesLabel->{"x","y","F"},ColorOutput->CMYKColor];
```



The last one is the organization center of the parabolic umbilic:

```
Plot3D[x^2 y+y^4,{x,-4,4},{y,-3,3}, AspectRatio->2,
BoxRatios->{1,1,2}, Boxed->False,PlotPoints->30,ViewPoint->{12,-5,2.},
Mesh->True,AxesLabel->{"x","y","F"}, DefaultFont->20,
```


ColorOutput->CMYKColor];



7.1 The fold

The fold catastrophe is the first in Thom's list of the seven elementary catastrophes. It is the universal unfolding of the singularity $f(x)=x^3$ and it has the equation $F(x,u)=x^3+u \cdot x$. We now define this equation in the Mathematica system and calculate the first and the second derivative with respect to x . (Please note the space between u and x for the multiplication!)

```
Remove["Global`*"]
```

```
F[x_,u_]:=x^3+u x
```

```
F[x,u]
```

$$u x + x^3$$

```
D1F[x_, u_] := Evaluate[D[F[x, u],x]]
```

```
D1F[x,u]
```

$$u + 3 x^2$$

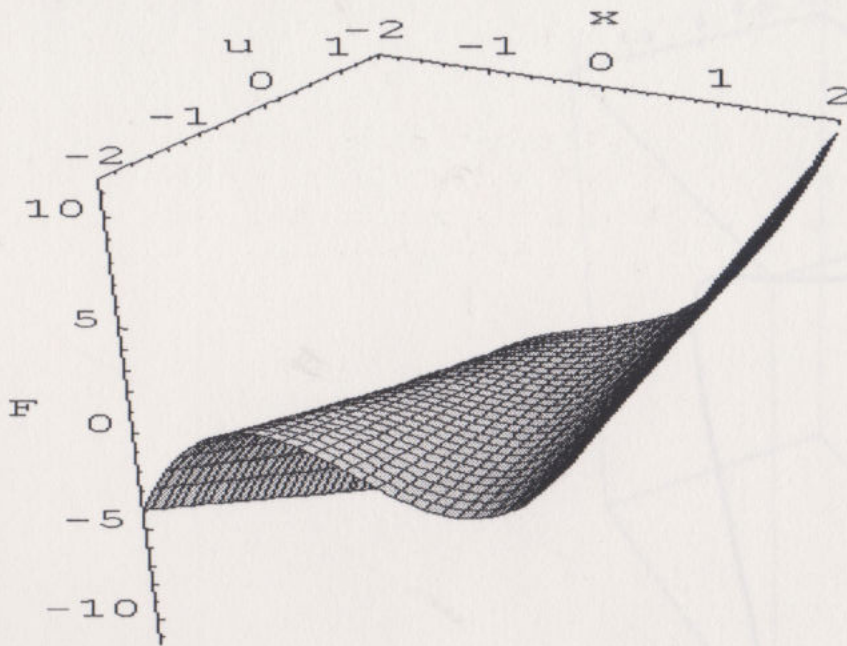
```
D2F[x_, u_] := Evaluate[D[D1F[x, u],x]]
```

```
D2F[x,u]
```

$$6 x$$

Now we want to draw $F(x,u)$ near the origin. We use the "Plot3D" statement of Mathematica.

```
Foldsurface = Plot3D[F[x, u], {x, -2, 2}, {u, -2, 1.5}, AspectRatio -> 2,
  BoxRatios -> {1, 1, 2}, Boxed -> False, PlotPoints -> 30,
  ViewPoint -> {1.3, -2.4, 2.}, Mesh -> True, AxesLabel -> {"x", "u", "F"},
  HiddenSurface -> True] ;
```

We are interested in the set of singular points of F relative to x , i.e. for those points where the first partial derivative with respect to x (the system variable) vanishes. The “Solve” statement solves the equation and the “Flatten” statement removes parentheses.

```
SingularPoints=Solve[D1F[x,u]==0,{u}]/Flatten
```

```
{u → -3 x2}
```

In order to be able to continue the calculation without giving the variable u a certain value, we put the solution into a variable named s .

```
s=u/.SingularPoints
```

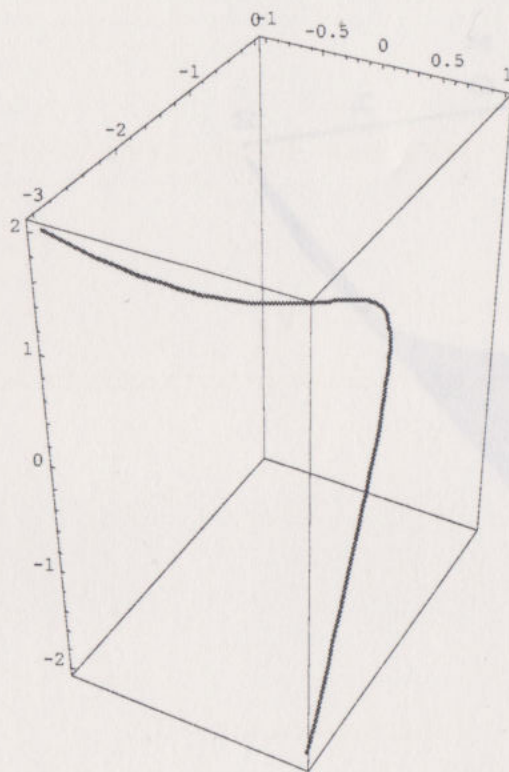
```
-3 x2
```

```
F[x,s]
```

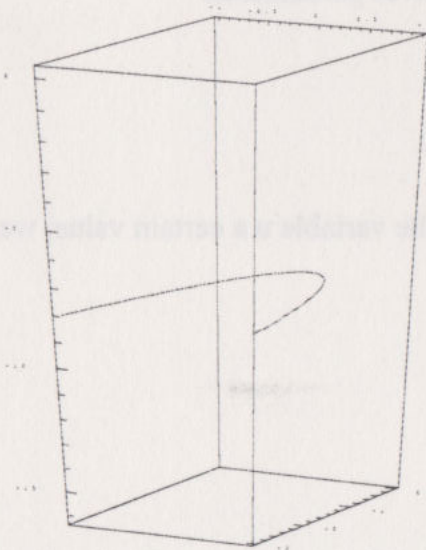
```
-2 x3
```

We draw the solution inserted into F , which gives a space curve. Its projection onto the x, u -plane is the parabola calculated above (Hint: For the reason of visibility, the plane of projection was moved a little bit downward, and a small value ϵ was added to the space curve. You are invited to do your own experiments with these values.)

```
SP=ParametricPlot3D [{x,s,F[x,s]},{Thickness[0.008],RGBColor[1, 0, 0]},{x,-1, 1}];
```



```
Projection = ParametricPlot3D[{x, s, -8,
    {Thickness[0.008], RGBColor[0, 0, 1]}}, {x, -1, 1}];
```



```
epsilon=0.01;
```

```
SPvisible = ParametricPlot3D[{x, s, F[x, s] + epsilon,
    {Thickness[0.008], RGBColor[1, 0, 0]}}, {x, -1, 1},
    DisplayFunction -> Identity];
```

We make a graphic for the projection of the parabola onto parameter space (u-axis) :

```
Projectionontou = ParametricPlot3D[{0, s, -8,
```



```
{Thickness[0.01], RGBColor[0, 1, 0]}}, {x, -1, 1},  
  DisplayFunction -> Identity};
```

The value of u , which belongs to the single degenerated critical point, is drawn as a red (only on PC display) point :

```
catastbegin =  
  Graphics3D[{RGBColor[1, 0, 0], PointSize[0.03], Point[{0, 0, -8}]}];
```

We make coordinate axes for the x, u, F directions. The starting point for the unit vectors is, for the reason of visibility, $(0,0,-8)$ instead $(0,0,0)$.

```
<<Graphics`PlotField3D`
```

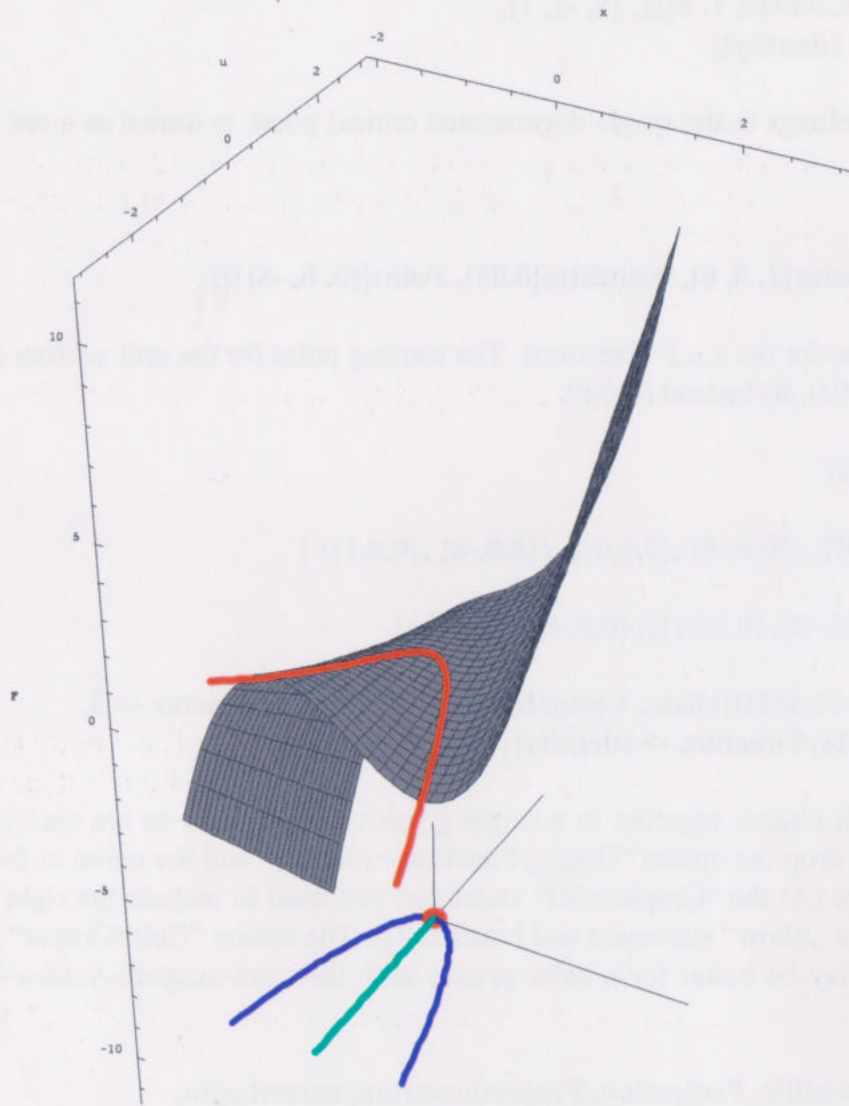
```
vlist = { {{0,0,-8}},{1,0,0}}, {{0,0,-8}},{0,1,0}}, {{0,0,-8}},{0,0,1}} }
```

```
{{{0,0,-8},{1,0,0}},{{0,0,-8},{0,1,0}},{{0,0,-8},{0,0,1}}}
```

```
across = ListPlotVectorField3D[vliste, VectorHeads -> False, ScaleFactor -> 3,  
  Axes -> False, DisplayFunction -> Identity];
```

Now we must bring all objects together in a single graphic. If you want to see each of the graphics alone, you may drop the option "DisplayFunction->Identity" and the colon in front of it in the upper statements (At the "Graphics3D" statement you need to include the right hand side of the statement in a „Show“ statement and brackets []). The option "ColorOutput" gives brighter colors, which may be better for a color-printer with the cyan-magenta-yellow-black color model.

```
Show[{Foldsurface, SPvisible, Projection, Projectionontou, catastbegin,  
  across},ViewPoint -> {1.3, -2.4, 2}, ColorOutput -> CMYKColor];
```



In the picture you can see $F(x,u)$ and axes in x,u,F directions. The curve, displayed in red on your monitor, joins the extremes of F and the blue parabola represents their x,u values. There is only one degenerated critical point of the fold (at the vertex of the Parabola), i.e. a critical point where the second derivative by x also vanishes. The two branches of the parabola in $x-u$ plane give the positions of the maximums res. the minimums of F . In the points characterizing a minimum, the system, having this kind of singularity, is stable, while in a maximum it is unstable. Projecting the parabola onto parameter space (u -axis), one gets the (on display) green line. For parameter values of u , that are negative, we are on the green line. There the system has a stable minimum (and an unstable maximum, not observable in nature). For parameter values $u > 0$ the system has no stability points at all. Interpreting the parameter u as time, the point $u=0$ (red point in the display graphic) is the beginning or the end of a stable system behavior. Here the system shows catastrophic behavior. The morphology of the fold is a "beginning" or "end".

7.2 The cusp

The cusp catastrophe is the universal unfolding of the function germ $f(x)=x^4$. Its equation is $F(x,u,v)=x^4+u \cdot x^2+v \cdot x$. One cannot draw such a function of three variables in 3-space. We may try to draw sections and this is what we will do later. First we want to look at its catastrophe surface, that is, the set of the singular points of F . Those are the points in (x,u,v) -space, where the partial derivative of F , with respect to the system variable x , is zero. It is, as we shall see, a surface in 3-space.

First we clear all variables in Mathematica, then we define F and its partial derivative $D1F$ with respect to x .

```
Remove["Global`*"]
```

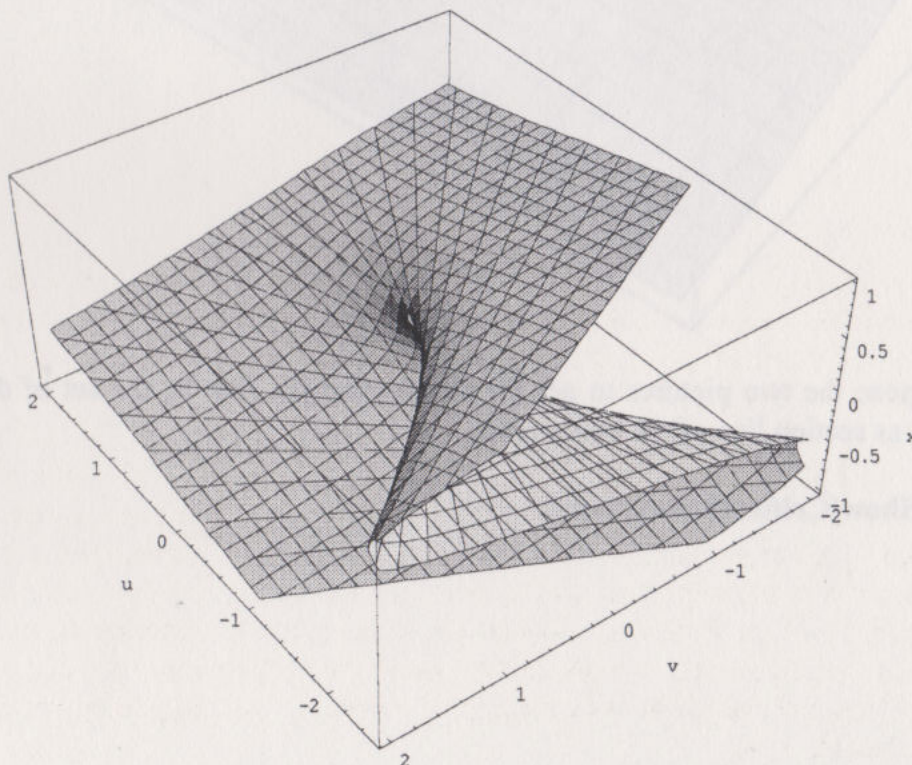
```
F[x_, u_, v_] := x^4 + u*x^2 + v*x
```

```
D1F[x_, u_, v_] := Evaluate[D[F[x, u, v], x]]
```

Using the „ContourPlot3D“ statement, we now draw the surface $D1F=0$. It is not necessary to write the „=0“ explicitly. If your PC is not very quick, you may need a little bit of patience for the following graphics:

```
<< "Graphics`Master`"
```

```
CriticalPoints = ContourPlot3D[D1F[x, u, v], {u, -2.5, 2}, {v, -2, 2}, {x, -1, 1},  
PlotPoints -> 6, ViewPoint -> {-2, 1.5, 2}, Axes -> True, AxesLabel -> {"u", "v", "x"},  
DefaultFont -> 14]
```

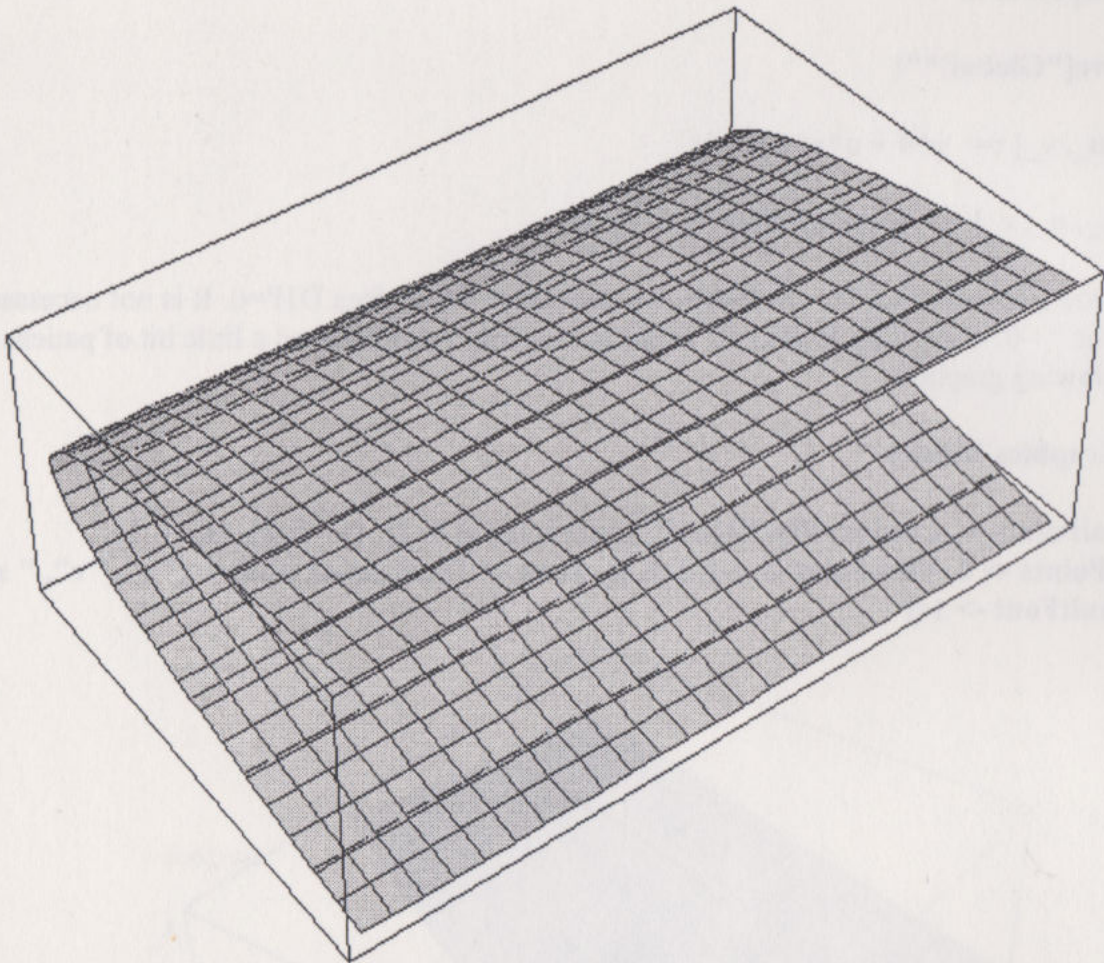


What you see in this picture is the catastrophe surface of the cusp catastrophe F . Often it is also called catastrophe manifold and should not be confused with the catastrophe set defined below.

The same drawing procedure works for the set of points (x,u,v) , where the second partial derivative of $F(x,u,v)$ with respect to x vanishes:

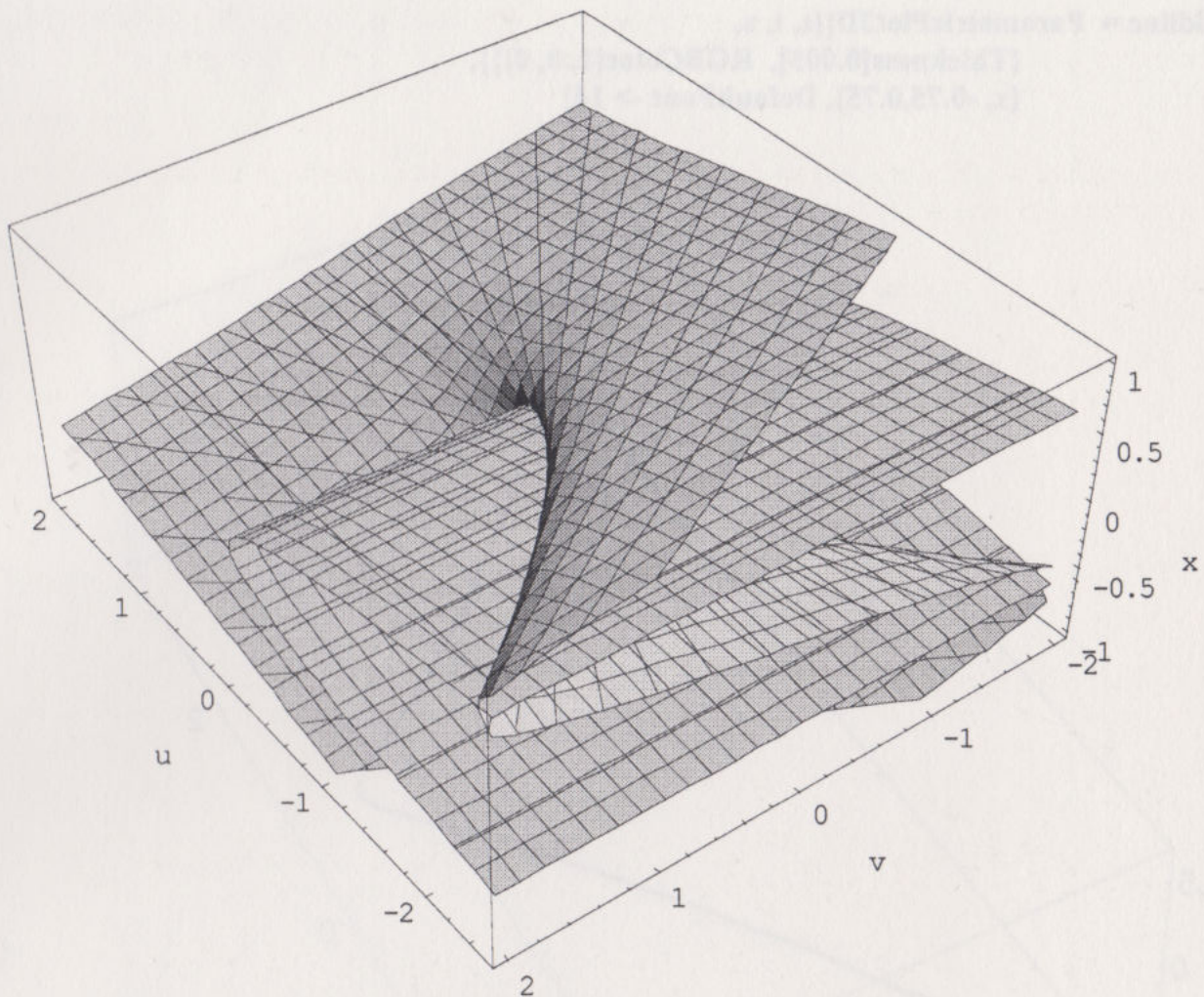
```
D2F[x_, u_, v_] := Evaluate[D[D1F[x, u, v], x]]
```

```
scnd = ContourPlot3D [D2F[x, u, v], {u, -2.5, 2}, {v, -2, 2},  
  {x, -1, 1}, PlotPoints -> 6, ViewPoint -> {-2., 1.5, 2}]
```



We now compose the two pictures to get the catastrophe set, that is, the set of degenerated critical points, as section line of the two graphs:

```
compose1 = Show[CriticalPoints, scnd]
```

Now we do the computation for the degenerated critical points (catastrophe set, which we here also call „fold line“). We solve the equations which are defining the fold line:

```
solu = Solve[{D1F[x, u, v] == 0, D2F[x, u, v] == 0}, {u, v}]
```

```
{{v -> 8x3, u -> -6x2}}
```

We remove the brackets:

```
solu1 = Flatten[solu]
```

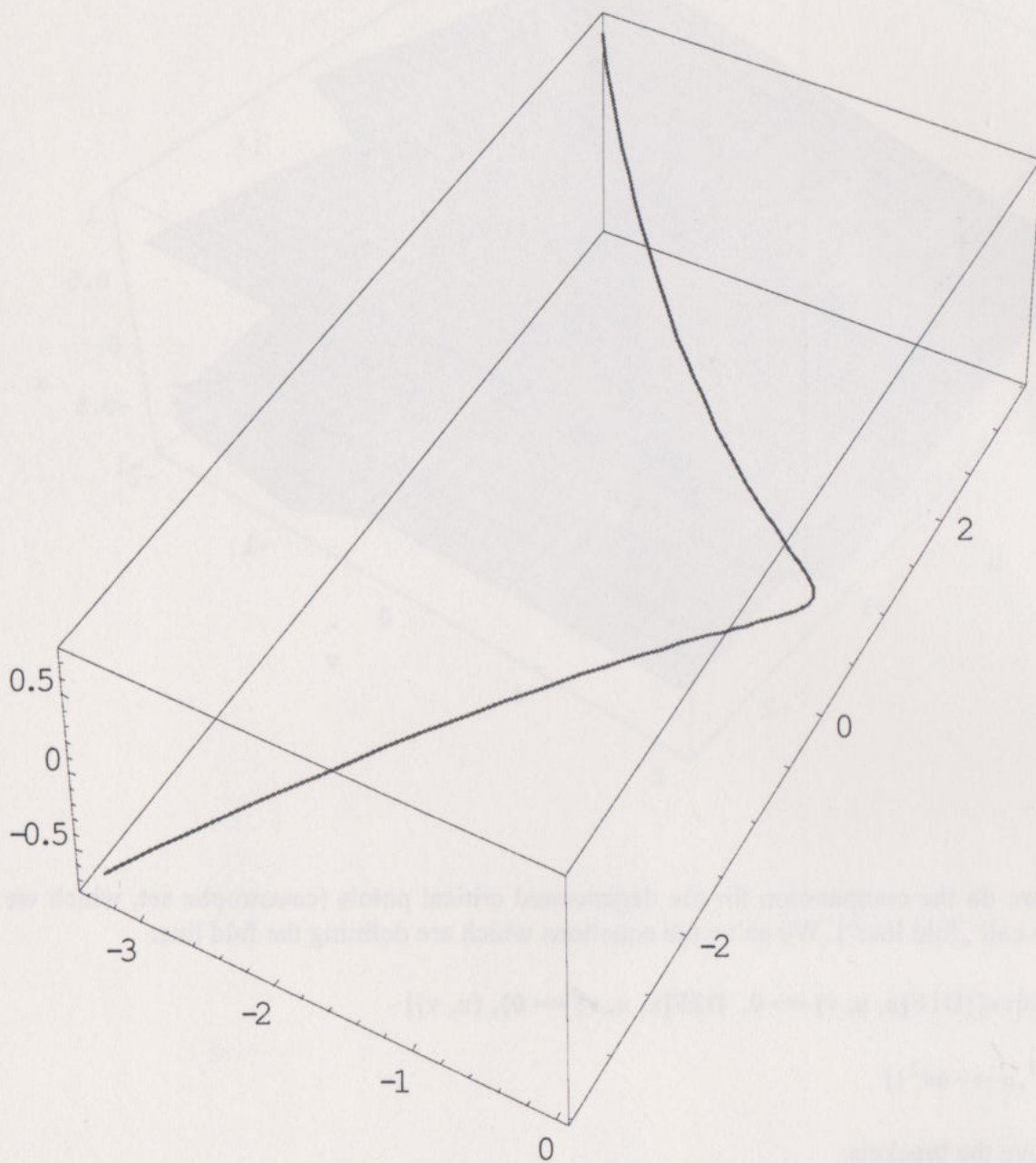
```
{v -> 8x3, u -> -6x2}
```

For the following computation we rename u and v or a short moment s and t:

```
{s, t} = {u, v} /. solu1
```

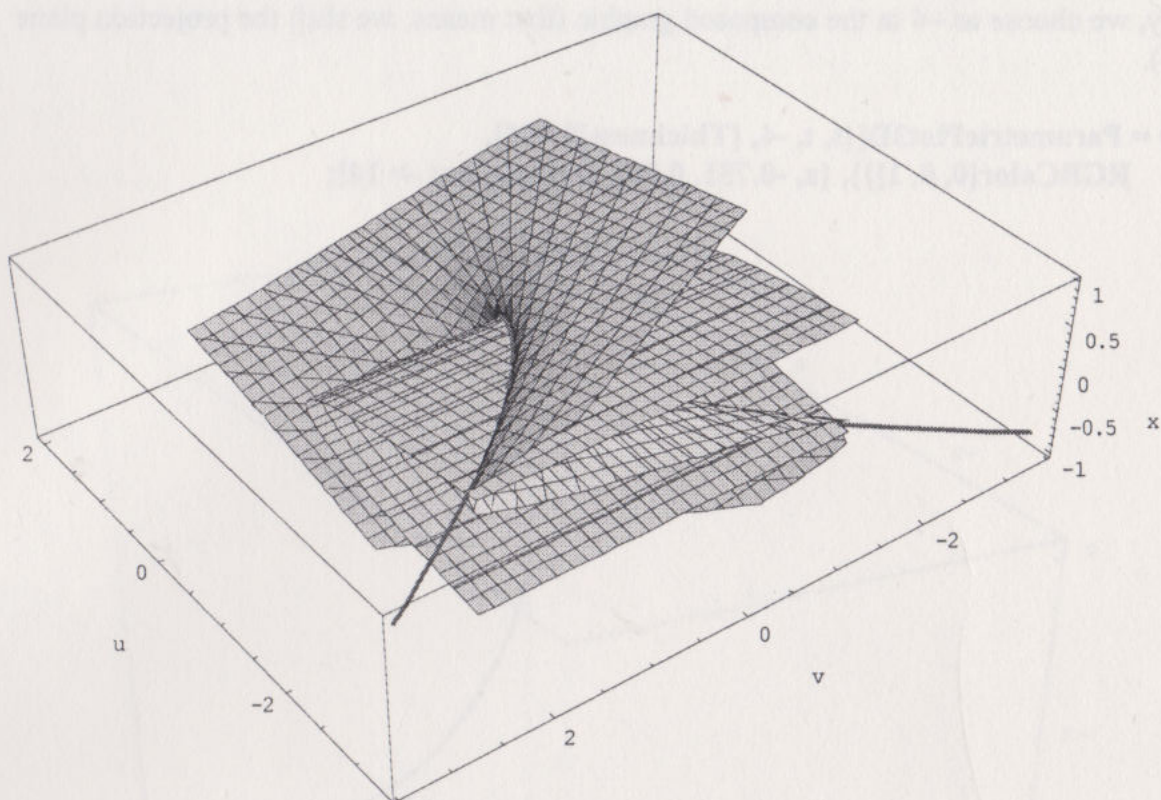
```
{-6x2, 8x3}
```

```
foldline = ParametricPlot3D[{s, t, x,
    {Thickness[0.005], RGBColor[1, 0, 0]}},
    {x, -0.75, 0.75}, DefaultFont -> 14]
```



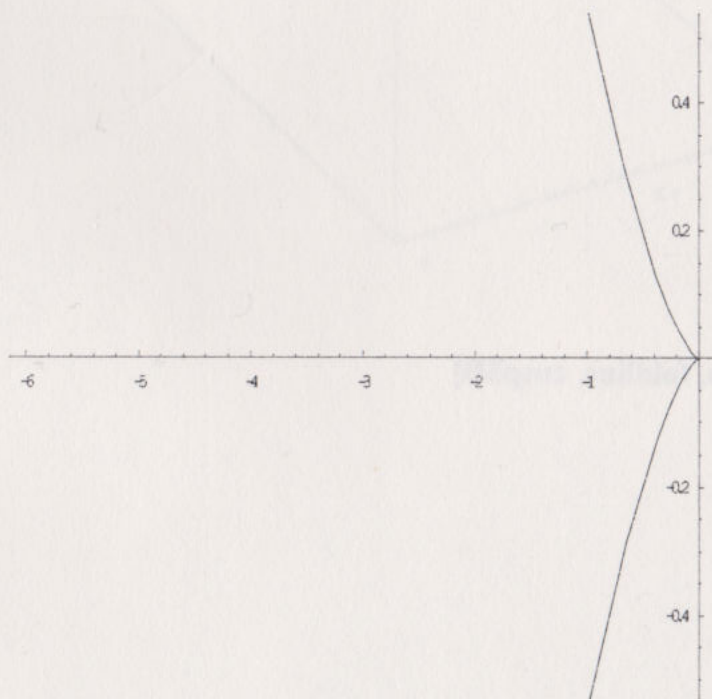
This is now shown together with both surfaces:

```
compose2 = Show[compose1, foldline]
```

Let's look at the projection of the fold line onto parameter space, it is, onto u - v -plane. This projected curve is called bifurcation set or cusp curve, from which this catastrophe has its name.

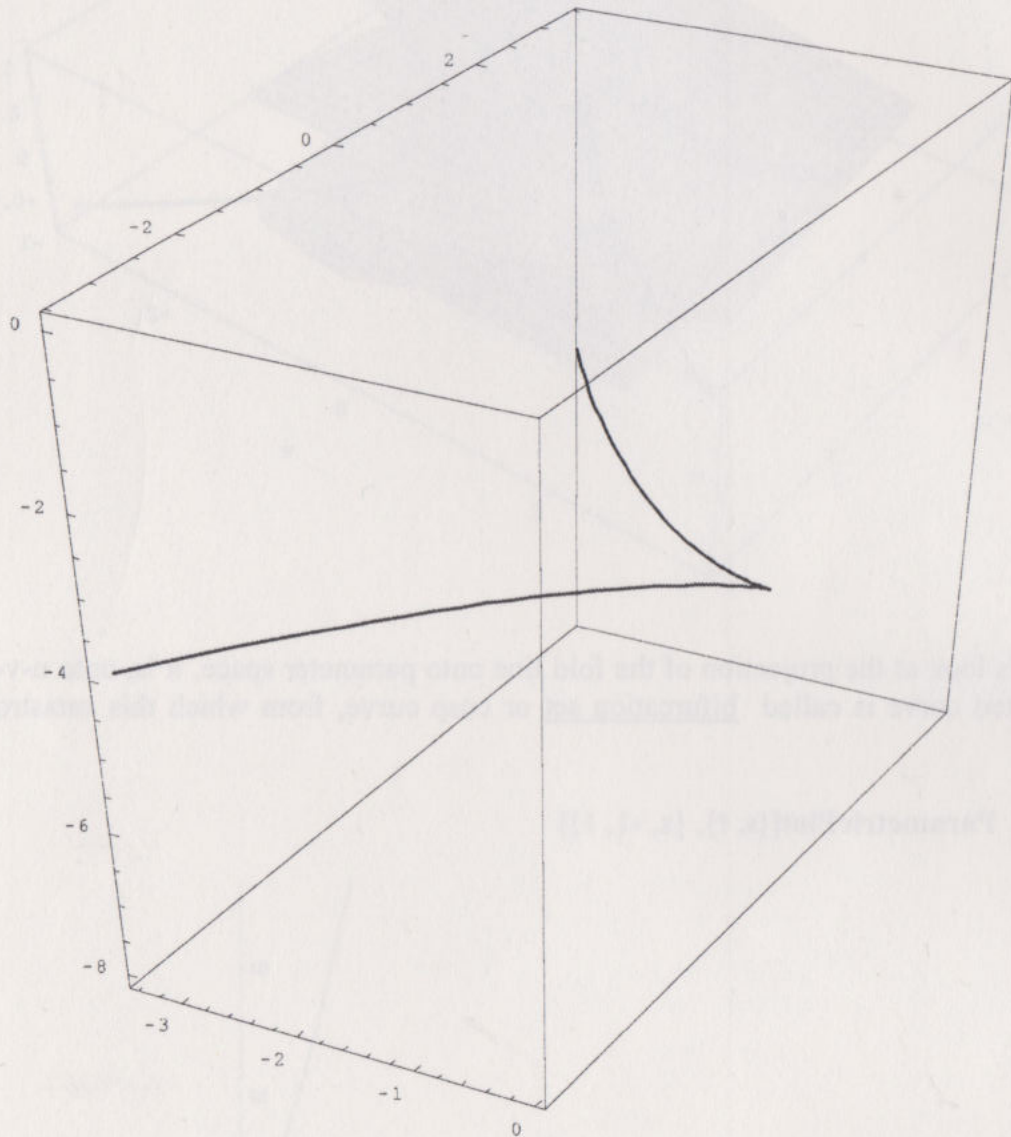
```
cusps = ParametricPlot[{s, t}, {x, -1, 1}]
```



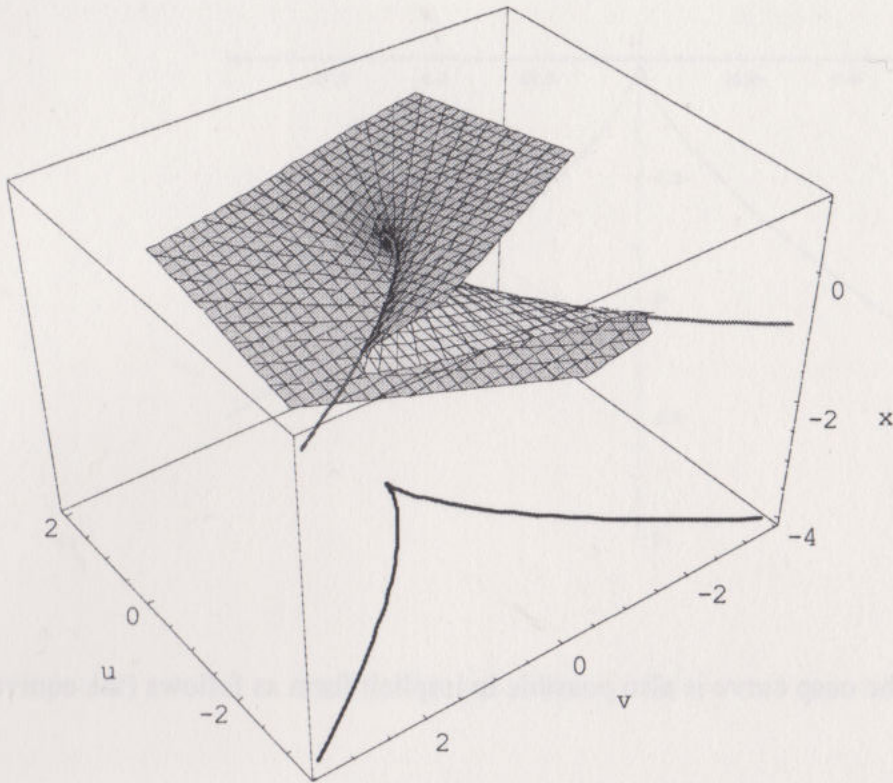
For a picture of that plane curve in 3-space we need a third component which, for reason of

visibility, we choose as -4 in the composed graphic (that means, we shift the projection plane to $x=-4$).

```
cuspid3D = ParametricPlot3D[{s, t, -4, {Thickness[0.005],  
    RGBColor[0, 0, 1]}}, {x, -0.751, 0.75}, DefaultFont -> 14];
```



```
pic2 = Show[CriticalPoints, foldline, cuspid3D]
```

Let's have a closer look at the bifurcation set, that is the projection of the set of degenerated critical points (fold line, catastrophe set) onto parameter space. It gives a decomposition of parameter space (v-u plane) into regions, over which the function F shows some characteristic features:

We clear the variables and load the Graphics`Master package

```
Clear[u,v,L]
```

```
Needs["Graphics`Master`"]
```

Next we calculate the u and v values, for which the function F has degenerated critical points. This is done by solving the following equations:

```
L=Solve[{4*x^3 + 2*u*x + v == 0, 12*x^2 + 2*u == 0}, {u, v}]
```

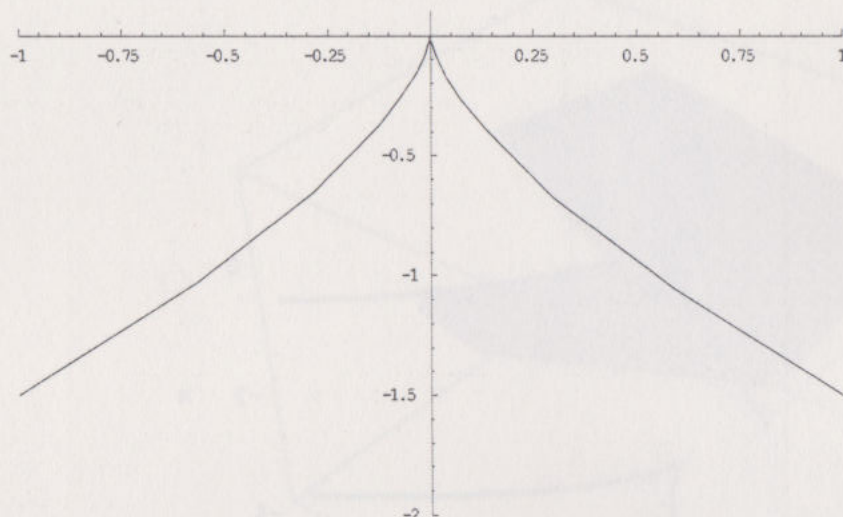
```
{{v -> 8 x^3, u -> -6 x^2}}
```

```
{s,t}={u,v}/.L//Flatten
```

```
{-6 x^2, 8 x^3}
```

We choose the "ParametricPlot" statement for drawing the solution curve:

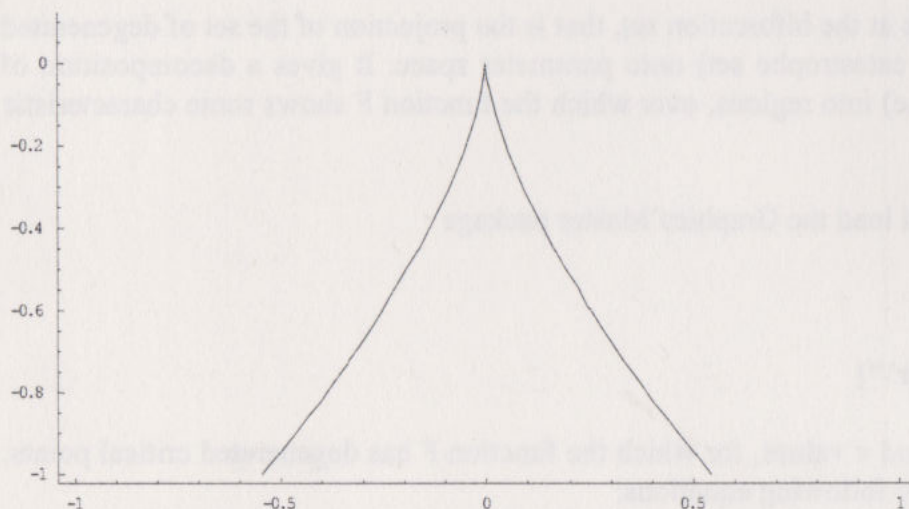
```
pic1=ParametricPlot[{t,s},{x,-1,1},PlotRange->{{-1,1},{-2,0.1}}]
```



Clear[u,v]

A representation of the cusp curve is also possible in implicit form as follows (the equivalence is easy to prove):

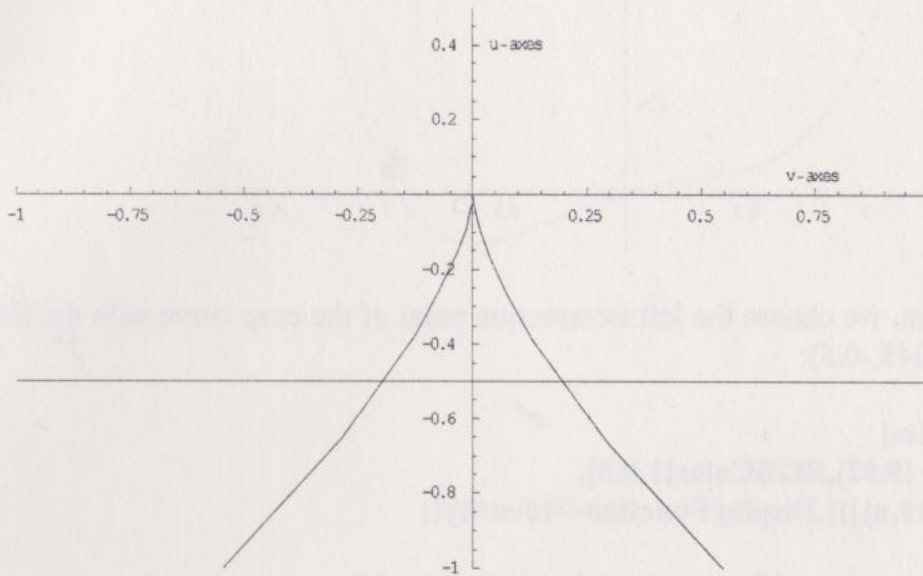
cuspl = ImplicitPlot[8*u^3 + 27*v^2 == 0, {v, -1, 1}, {u, 0.1, -1}]



The following input lines in Mathematica all have the same purpose: They generate some graphics in one single graphic array, which make clear, how the sections of $F(x,u,v)$ look like, if for u and v certain values of parameter space are taken. Of special interest are the cases, where the points of parameter space are chosen to lie on the cusp curve (bifurcation set). There the function F has (with respect to x) a saddle and a minimum. Why just this configuration is so important, we will show below. The positions of the points (on the display in color) and the corresponding sections (in the same color) are shown in a single total graphic, where the bifurcation set is in the middle of the picture. Please note, how the function $x^4 + u \cdot x^2 + v \cdot x$ changes its form, depending on u and v . The reader, who doesn't want to write all input lines, may just have a look on the resulting graphics. He may step over these input lines without problems and continue his inputs after the graphics array and the following text in this book.

Clear[u,v]

We want to move in parameter space (v-u plane) along the line $u=-0.5$ from negative v values to positive v values, that means, we move along the line, which cuts the cusp in the following picture.



First of all, we calculate the intersection points of the curve and the line, that we have chosen arbitrarily ($u=-0.5$):

u=-0.5;

Solve[8*u^3 + 27*v^2 == 0, v]

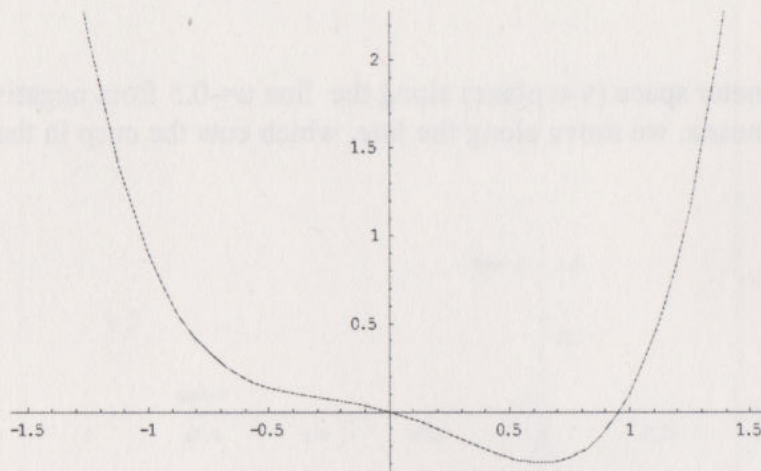
{{v -> -0.19245}, {v -> 0.19245}}

The first point, where we want to study the behavior of the function F, may be left of the value $v=-0.19245$, for example at $v=-0.4$

**P1=Show[Graphics[
 {PointSize [0.02],RGBColor[0.5,0.0,0.2],
 Point[{-0.4,u}]],DisplayFunction->Identity];**

The section of F belonging to these parameter values $(v,u)=(-0.4,-0.5)$ has the following form:

**pic1=Plot[x^4-0.5 x^2-0.4 x,{x,-1.5,1.5},
 PlotStyle->{RGBColor[0.5,0.0,0.2]}];**

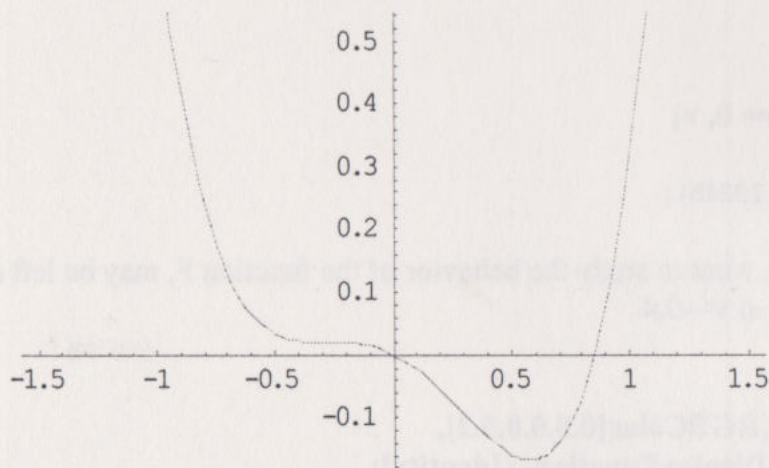


As the second point, we choose the left intersection point of the cusp curve with the line, that is the point $(-0.19245, -0.5)$:

```
P2=Show[Graphics[
  {PointSize [0.02],RGBColor[1,0,0],
   Point[{-0.19,u}]}],DisplayFunction->Identity];
```

The corresponding section of F has one minimum and a saddle.

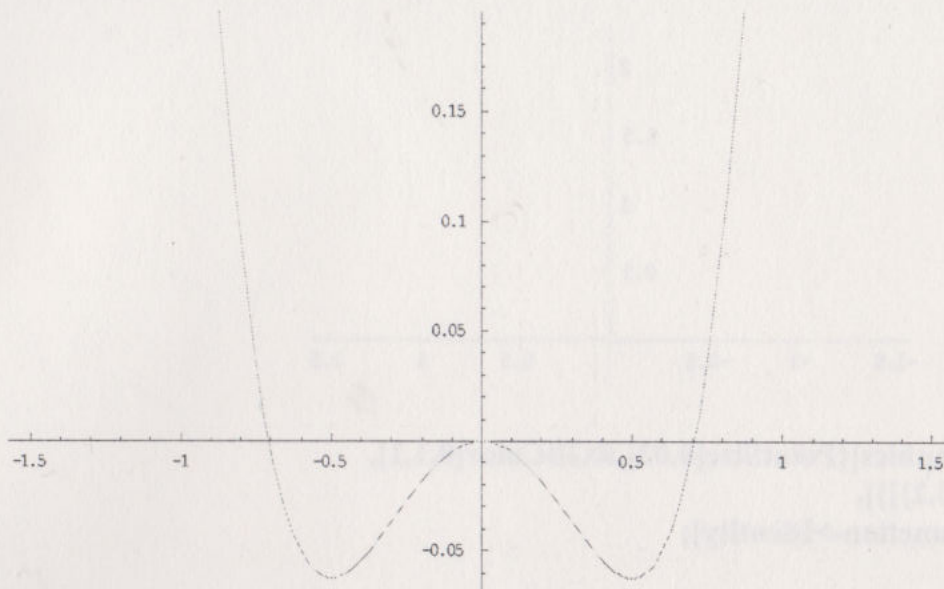
```
pic2=Plot[x^4-0.5 x^2-0.19245 x,{x,-1.5,1.5},
  PlotStyle->{RGBColor[1,0,0]}];
```



We choose still more points in parameter space. You can find their coordinates in the corresponding „Point“-statements. These points are shown later in the summary graphic.

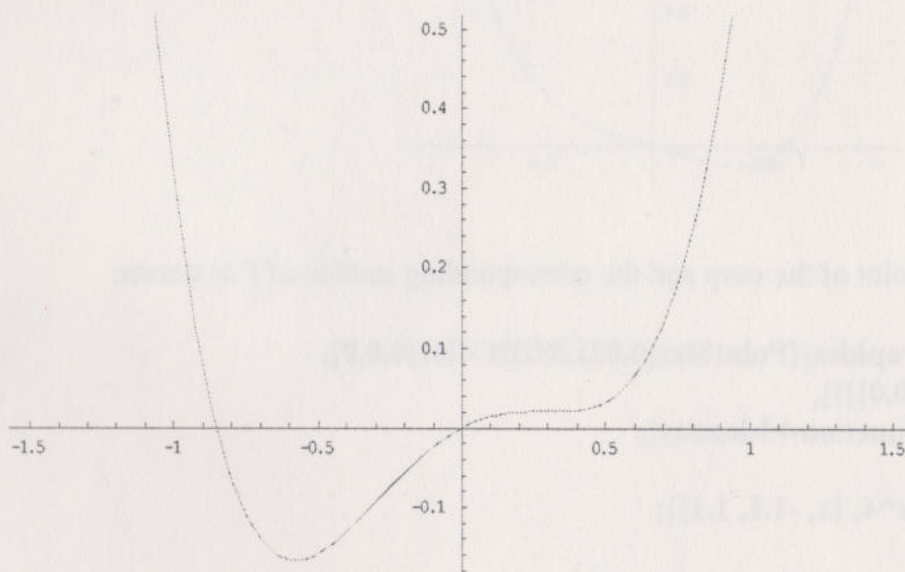
```
P3=Show[Graphics[
  {PointSize [0.02],RGBColor[0,0.5,0.8],
   Point[{0,u}]}],DisplayFunction->Identity];
```

```
pic3=Plot[x^4-0.5 x^2+0 x,{x,-1.5,1.5},
  PlotStyle->{RGBColor[0,0.5,0.8]}];
```

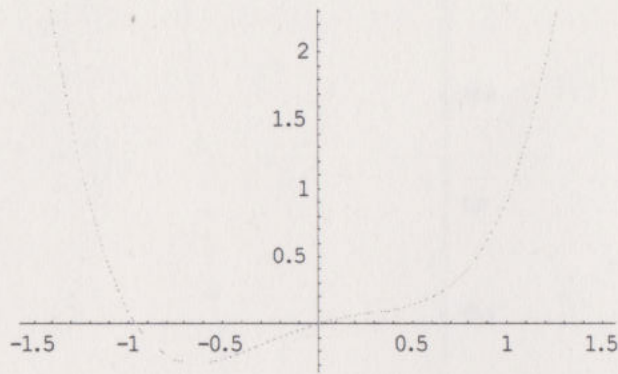
```
P4=Show[Graphics[
  {PointSize [0.02],RGBColor[0.5,0.2,0.3],
   Point[{0.19,u}]}],DisplayFunction->Identity];
```

```
pic4=Plot[x^4-0.5 x^2+0.19245 x,{x,-1.5,1.5},
  PlotStyle->{RGBColor[0.5,0.2,0.3]}];
```



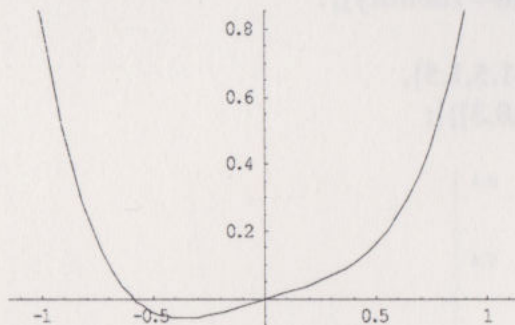
```
P5=Show[Graphics[
  {PointSize [0.02],RGBColor[0,1,0],
   Point[{0.4,u}]}],DisplayFunction->Identity];
```

```
pic5=Plot[x^4-0.5 x^2+0.4 x,{x,-1.5,1.5},
  PlotStyle->{RGBColor[0,1,0]}];
```



```
P6=Show[Graphics[{PointSize[0.02],RGBColor[0,1,1],
Point[{0,0.2}]}],
DisplayFunction->Identity];
```

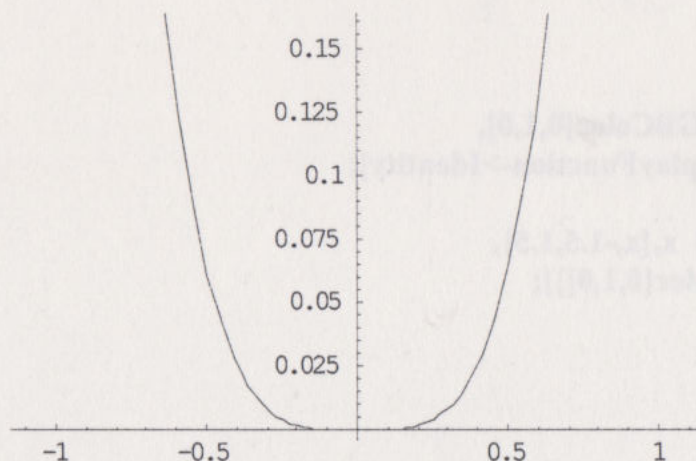
```
pic6= Plot[x^4 + 0*x^2 + 0.2*x, {x, -1.1, 1.1},
PlotStyle -> {RGBColor[0, 1, 1]}];
```



Now the origin point of the cusp and the corresponding section of F is drawn:

```
Origo=Show[Graphics[{PointSize[0.02],RGBColor[0,0,0],
Point[{0,0.0}]}],
DisplayFunction->Identity];
```

```
picorigo = Plot[x^4, {x, -1.1, 1.1}];
```



The following is the input for the labels of the graphic:

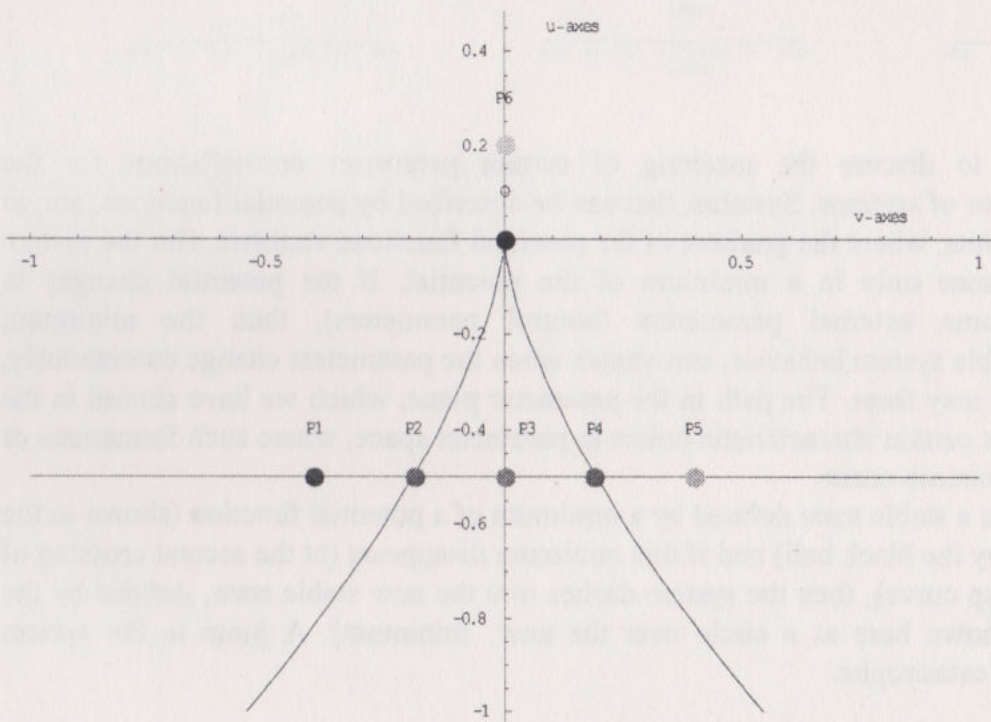

```
txt1=Show[Graphics[
  {Text[" ",{-0.5,u-0.1}], Text[" ",{0.5,0.4}],Text["O",{0,0+0.1}],
  Text["P6",{0,0.2+0.1}],Text["P1",{-0.4,u+0.1}],
  Text["P2",{-0.19,u+0.1}],Text["P3",{0.05,u+0.1}],
  Text["P4",{0.19,u+0.1}], Text["P5",{0.4,u+0.1}}],
  DefaultFont->30,DisplayFunction->Identity];
```

```
axeslab=Show[Graphics[{Text["u-axes",{0.15,0.45}],
  Text["v-axes",{0.8,0.05}}],
  DefaultFont->30,DisplayFunction->Identity];
```

```
line=Show[ Graphics[
  {Thickness [0.003],RGBColor[0,0,1],
  Line[{{-1,-0.5},{1,-0.5}}]},DisplayFunction->Identity];
```

Now we present the cusp together with the points P1 to P6 in a single graphic. The line from P1 to P6 represents, as already mentioned, a path in the parameter plane, which we have chosen to be able to observe the influence of the changes in the parameters of F. In the points P1, P2, etc. of this line, the form of the function F, belonging to that parameter values, is shown in the same color as the points.

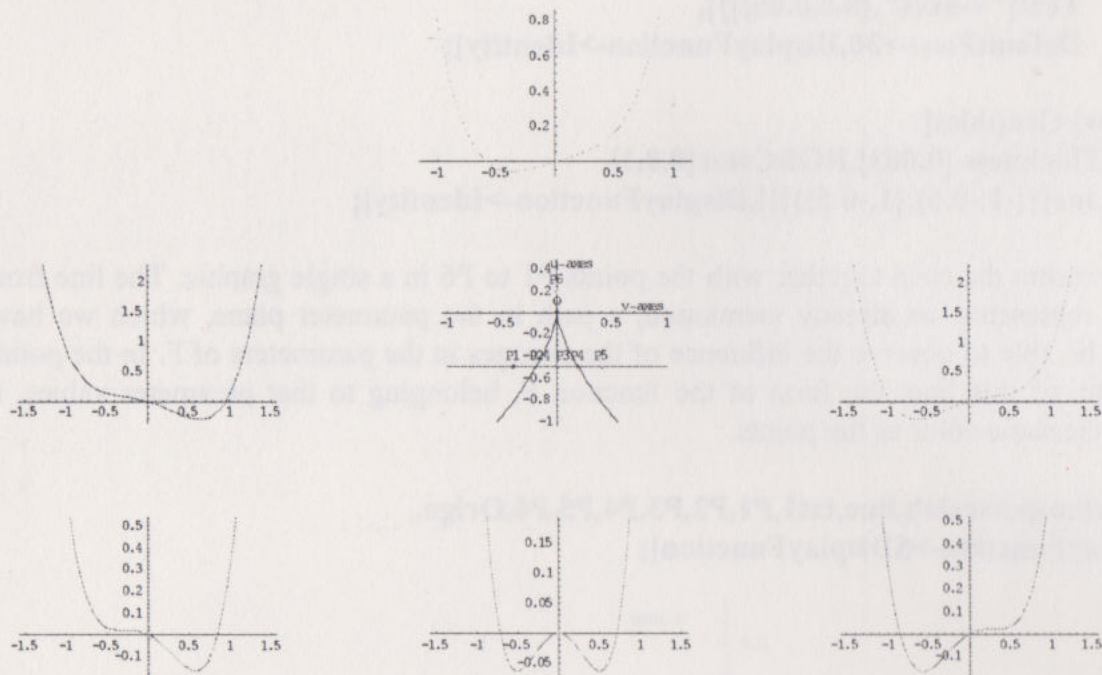
```
res=Show[cusp,axeslab,line,txt1,P1,P2,P3,P4,P5,P6,Origo,
  DisplayFunction->$DisplayFunction];
```



```
empt=Show[Graphics[{PointSize[0.002],RGBColor[0,1,1],
  Point[{0,0.2}}], DisplayFunction->Identity];
```

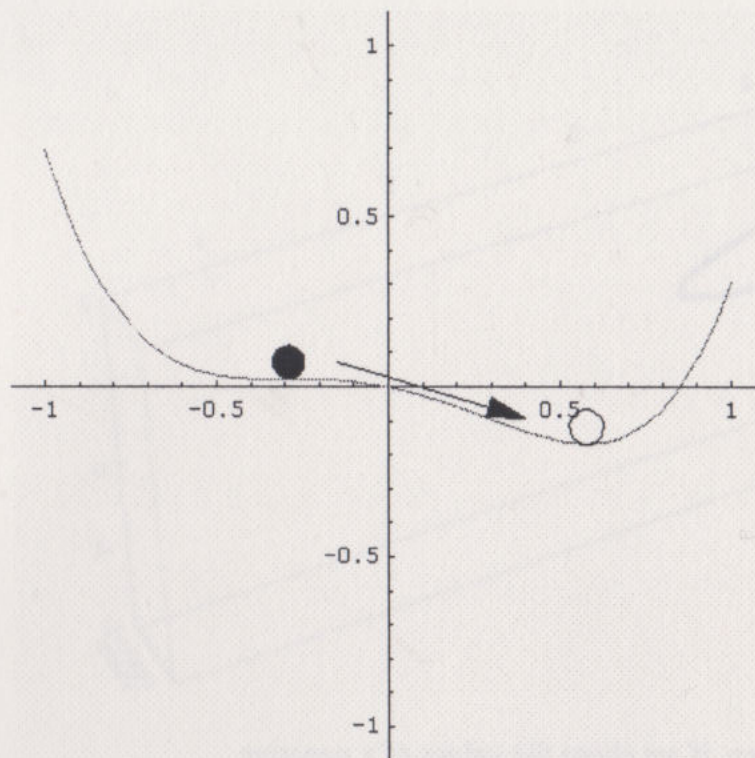
The statement above is used for the completion of the graphic array with an empty fill-in instead of a picture. Now we draw the complete graphic array showing the bifurcation set in the middle. Round about there are shown the graphs of the function F belonging to the points P1-P6 in parameter plane.

```
Show[GraphicsArray[
  {{empt,pic6,empt},{pic1,res,pic5},{pic2,pic3,pic4}},
  GraphicsSpacing->.5]];
```



We now want to discuss the meaning of certain parameter constellations for the catastrophic behavior of systems. Systems, that can be described by potential functions, are in balance at those points, where the gradient of the potential functions vanishes. But the system is in a stable balance only in a minimum of the potential. If the potential changes in dependence of some external parameters (control parameters), then the minimum, characterizing a stable system behavior, can vanish when the parameters change continuously, or a new minimum may form. The path in the parameter plane, which we have chosen in the last example, shows certain characteristic points in parameter space, where such formations or disappearances of minima occur.

If the system is in a stable state defined by a minimum of a potential function (shown in the following graphic by the black ball) and if this minimum disappears (at the second crossing of the line and the cusp curve), then the system dashes into the new stable state, defined by the other minimum (shown here as a circle over the new minimum). A jump in the system behavior occurs - a catastrophe.



In order to find more properties of the cusp catastrophe, we want to draw the surface of singularities (catastrophe surface) of F together with several sections. If, for example, we hold v at the value 0.5, then we have the following calculation for a section:

```
Clear[u]
```

```
vconst = 0.5;
```

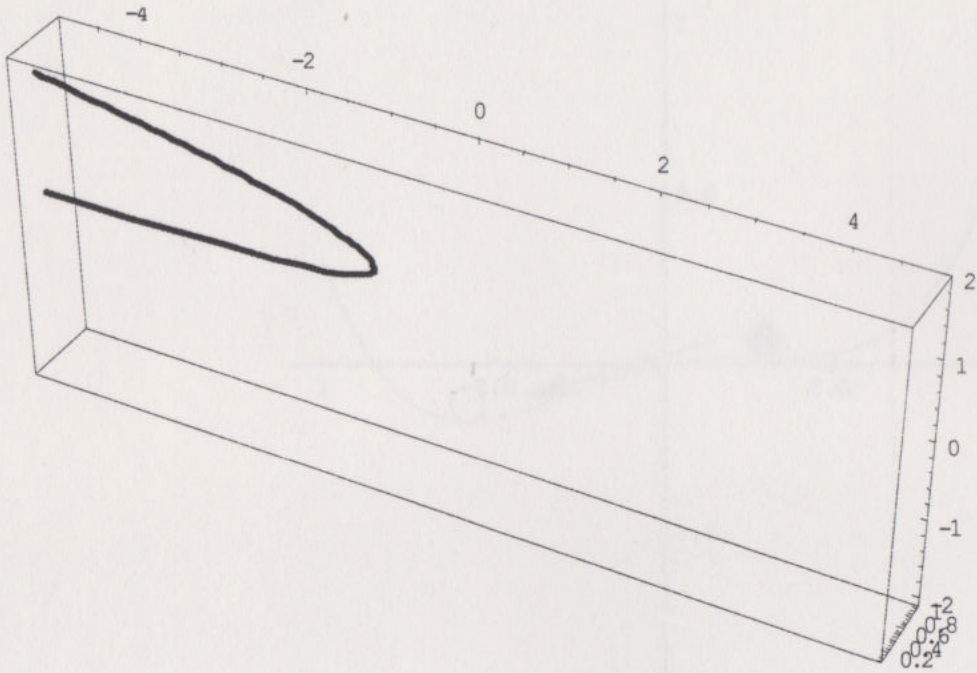
```
solu2 = Solve[D1F[x, u, vconst] == 0, {u}]
```

$$\left\{ \left\{ u \rightarrow -\frac{0.5 (0.5 + 4. x^3)}{x} \right\} \right\}$$

```
{u1}=u/.solu2//Flatten
```

$$\left\{ -\frac{0.5 (0.5 + 4. x^3)}{x} \right\}$$

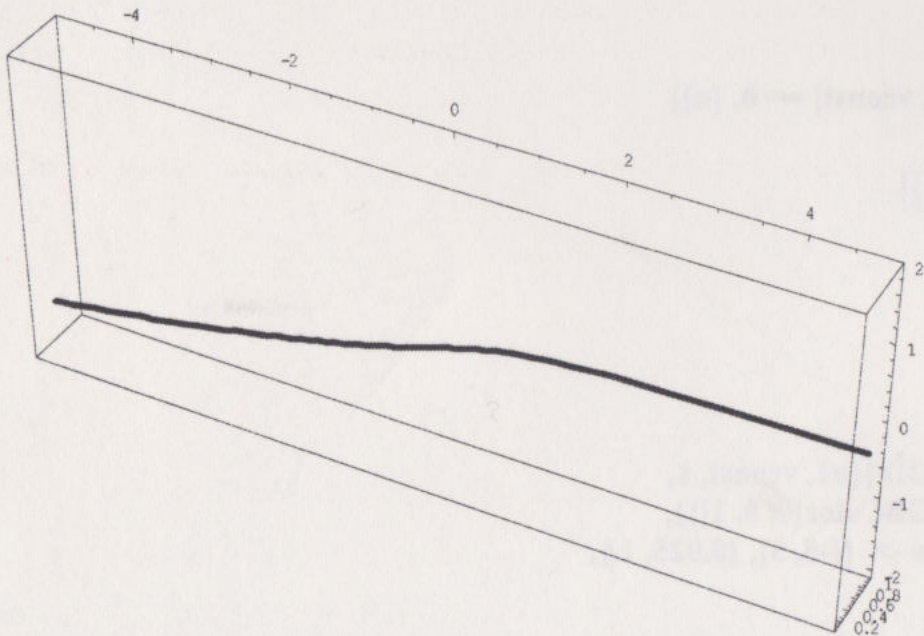
```
sec1v = ParametricPlot3D [{u1, vconst, x},
  {Thickness[0.008], RGBColor[0, 0, 1]}],
{x, 0.01, 2}, PlotRange -> {{-5, 5}, {0.025, 1.},
  {-2.1, 2.}}
```



Another section is given, if we chose the values of x negative

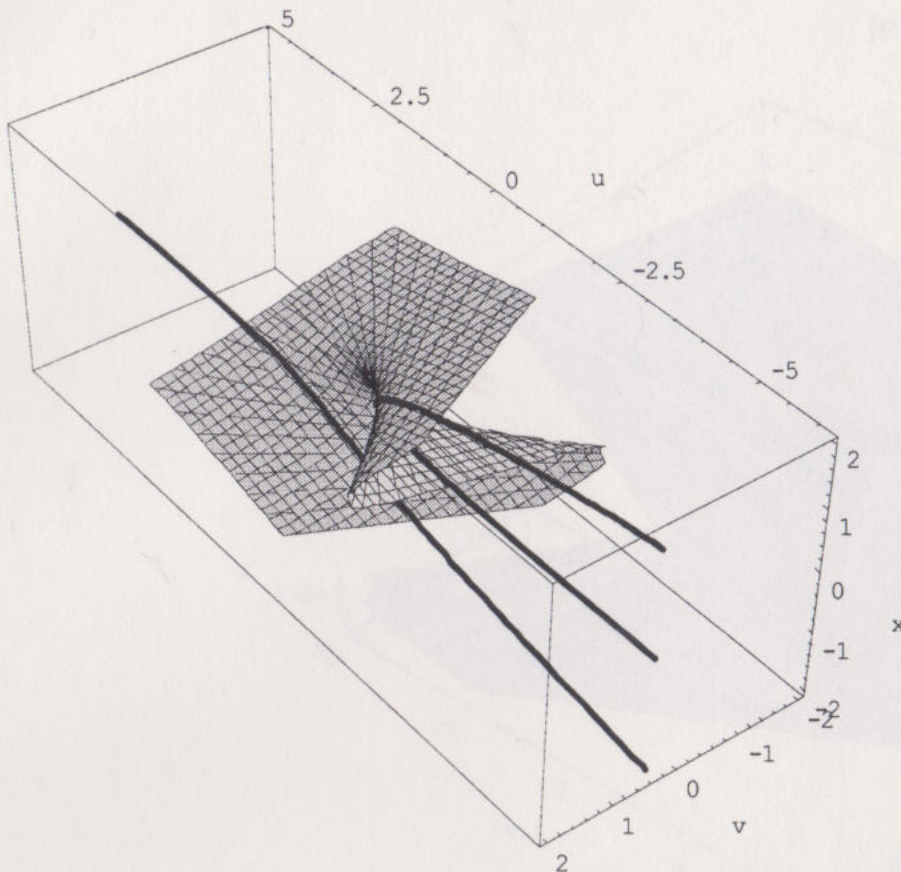
eps = 0.01;

```
sec2v = ParametricPlot3D[{u1, vconst, x + eps,
  {Thickness[0.008], RGBColor[0, 0, 1]}},
  {x, -2, -0.01}, PlotRange -> {{-5, 5}, {0.025, 1.}, {-2.1, 2.}}
```



Together, they give the following picture, which shows a typical property of the cusp: bifurcation

Show[CriticalPoints, sec1v, sec2v]



Going along the (blue) curve on the surface from positive parameter values u to negative values, the number of solutions of the equation $\partial_x F(x, u, v) = 0$ increases from one to three, the section along the curve suddenly has three branches instead of one. The outer branches represent (stable) minima of the function F , the stable equilibria of the system. The inner branch represents (unstable) maxima.

Another feature of the cusp is to be seen, if we now choose a section for a constant value of u , for example $u = -1$.

```
Clear[v]
```

```
uconst = -1;
```

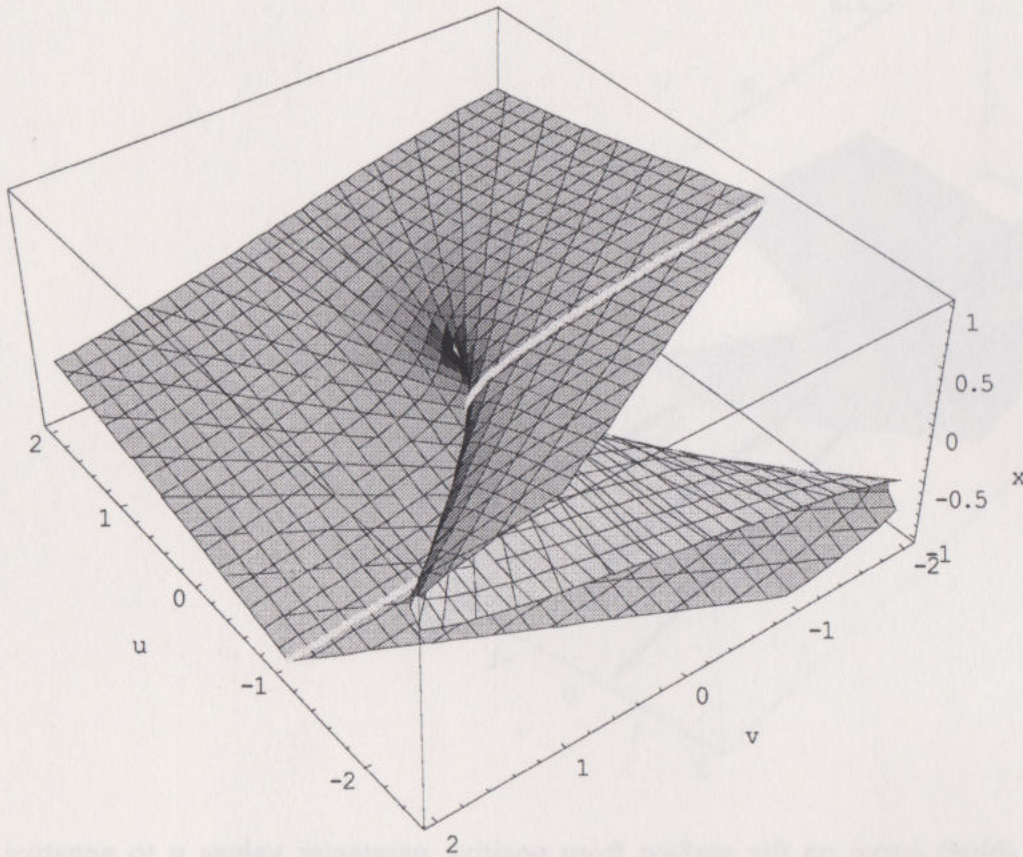
```
solu2 = Solve[D1F[x, uconst, v] == 0, {v}]
```

```
{v1}=v/.solu2//Flatten
```

```
{-2 (-x + 2 x^3) }
```

```
sec1u = ParametricPlot3D[{uconst, v1, x + eps,  
  {Thickness[0.008], RGBColor[1, 1, 0]}}, {x, -1, 1},  
  PlotRange -> {{-5, 5}, {-1, 1.}, {-2.1, 2.}},  
  DisplayFunction->Identity]
```

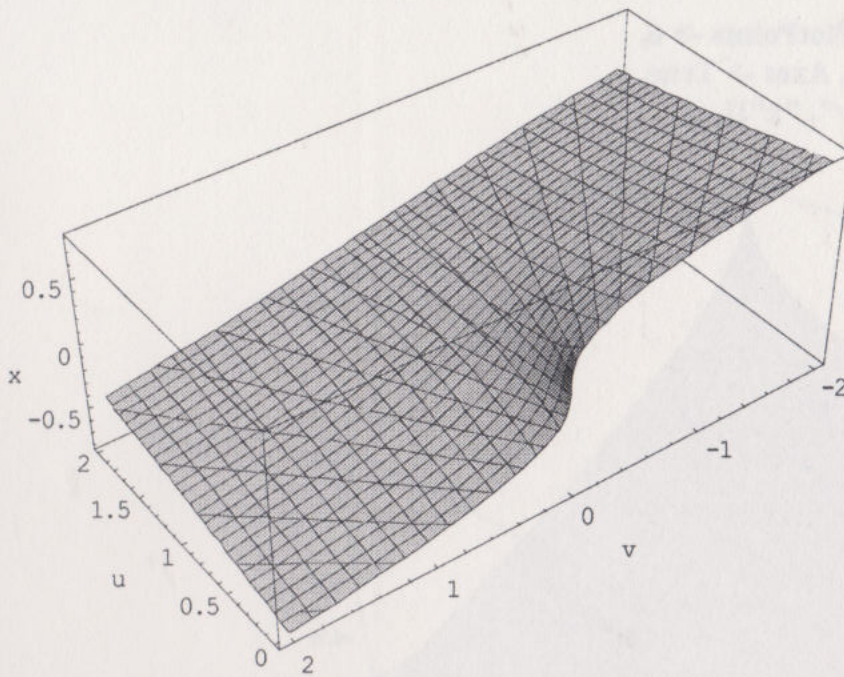

Show[CriticalPoints, sec1u]



This section shows a property of the cusp, that is called hysteresis. It occurs in different phenomena in physics, for example in magnetism. In our models, we understand by this property the following: Going along the curve (displayed in yellow) on the surface, the system, having this catastrophe manifold, may jump at the borders of the fold in its behavior to the lower (upper) leaf of the surface. These jumps do not happen at the same places when jumping goes from the upper leaf to the lower leaf or vice versa.

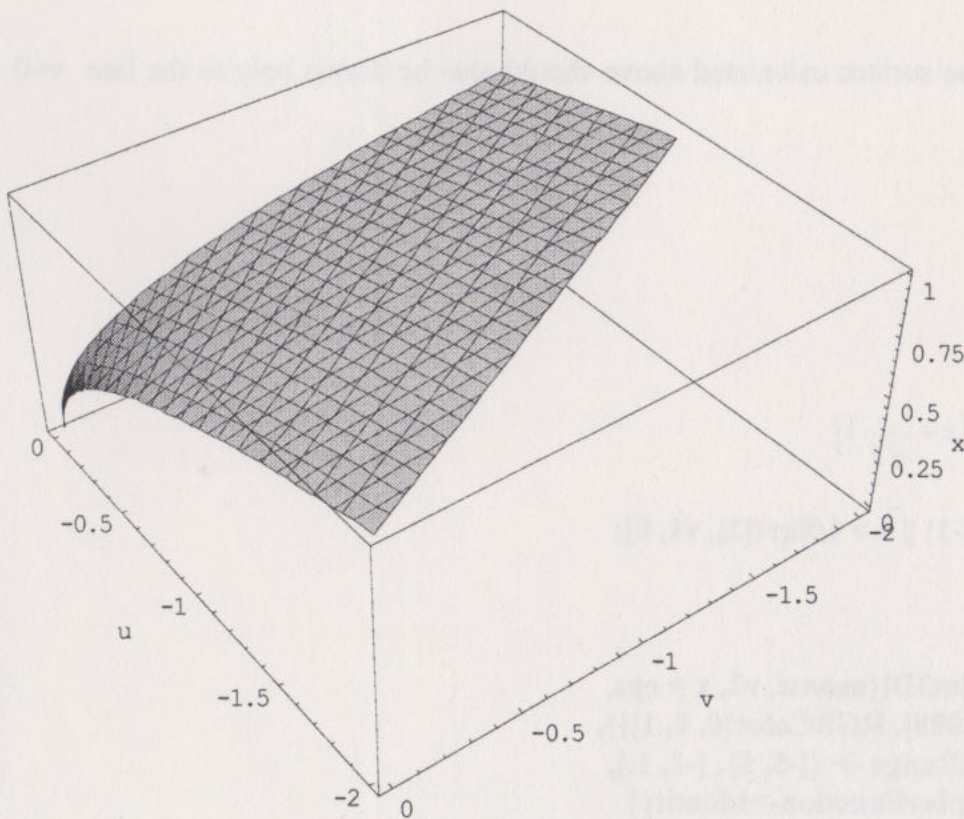
These jumps in the system behavior need not necessarily happen at the border of the fold („perfect-delay“ rule), but they may also happen over curves in parameter space (at lines, if the cusp is in normal form), which lie within the cusp curve in that plane („shock curves“, see also chapter 8). This rule for jumping of systems is called “Maxwell-convention“. To demonstrate graphically this fact, we need some pieces of the catastrophe surface, which we now calculate and the compose to one single graphics. First we draw that part of the surface, that has no fold:

```
C1 = ContourPlot3D[D1F[x, u, v], {u, 0, 2}, {v, -2, 2}, {x, -1, 1},  
  PlotPoints -> 6, ViewPoint -> {-2, 1.5, 2}, Axes -> True,  
  AxesLabel -> {"u", "v", "x"}, DefaultFont -> 14]
```

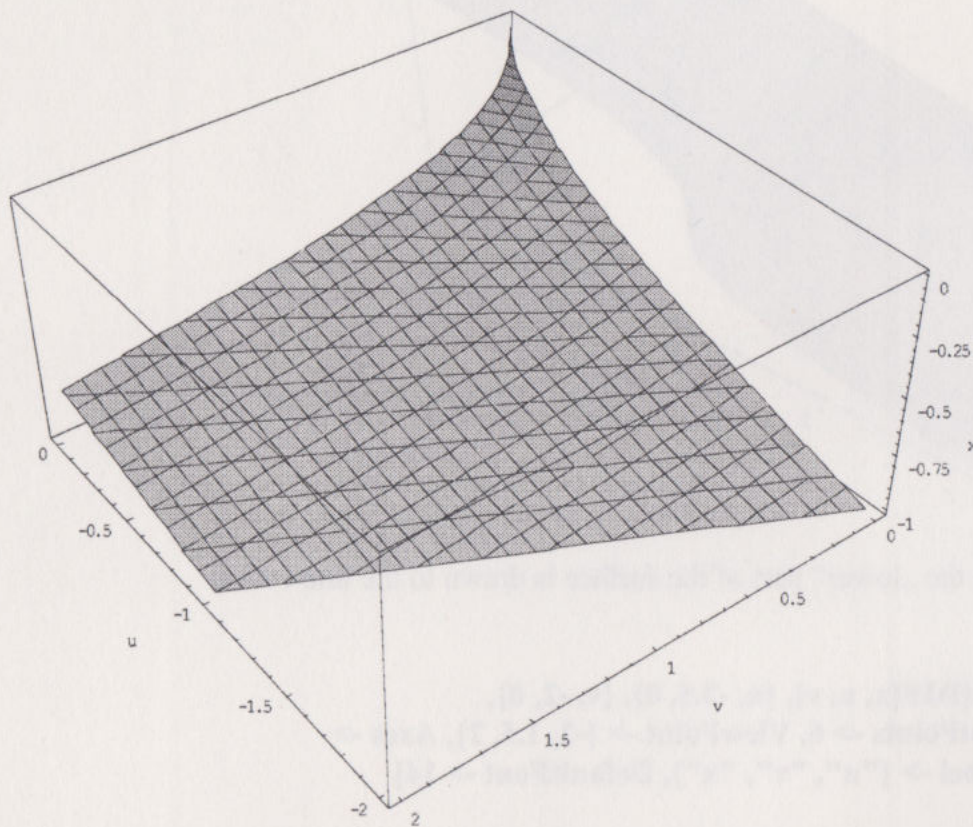
Now the „upper“ then the „lower“ part of the surface is drawn to the line $v=0$ in parameter plane.

```
C2 = ContourPlot3D[D1F[x, u, v], {u, -2.5, 0}, {v, -2, 0},  
{x, 0.01, 1}, PlotPoints -> 6, ViewPoint -> {-2, 1.5, 2}, Axes ->  
True, AxesLabel -> {"u", "v", "x"}, DefaultFont -> 14]
```



```
C3 = ContourPlot3D[D1F[x, u, v], {u, -2.5, 0},
```

```
{v, 0, 2}, {x, -1, -0.001}, PlotPoints -> 6,
ViewPoint -> {-2, 1.5, 2}, Axes -> True,
AxesLabel -> {"u", "v", "x"}}
```



The space curve on the surface calculated above should also be drawn only to the line $v=0$ and then be cut off.

v1

$$-2(-x + 2x^3)$$

n = Solve[v1 == 0, x]

$$\left\{ \{x \rightarrow 0\}, \left\{ x \rightarrow -\frac{1}{\sqrt{2}} \right\}, \left\{ x \rightarrow \frac{1}{\sqrt{2}} \right\} \right\}$$

v2 = If[x < -Sqrt[2]^(-1) || x > 1/Sqrt[2], v1, 0];

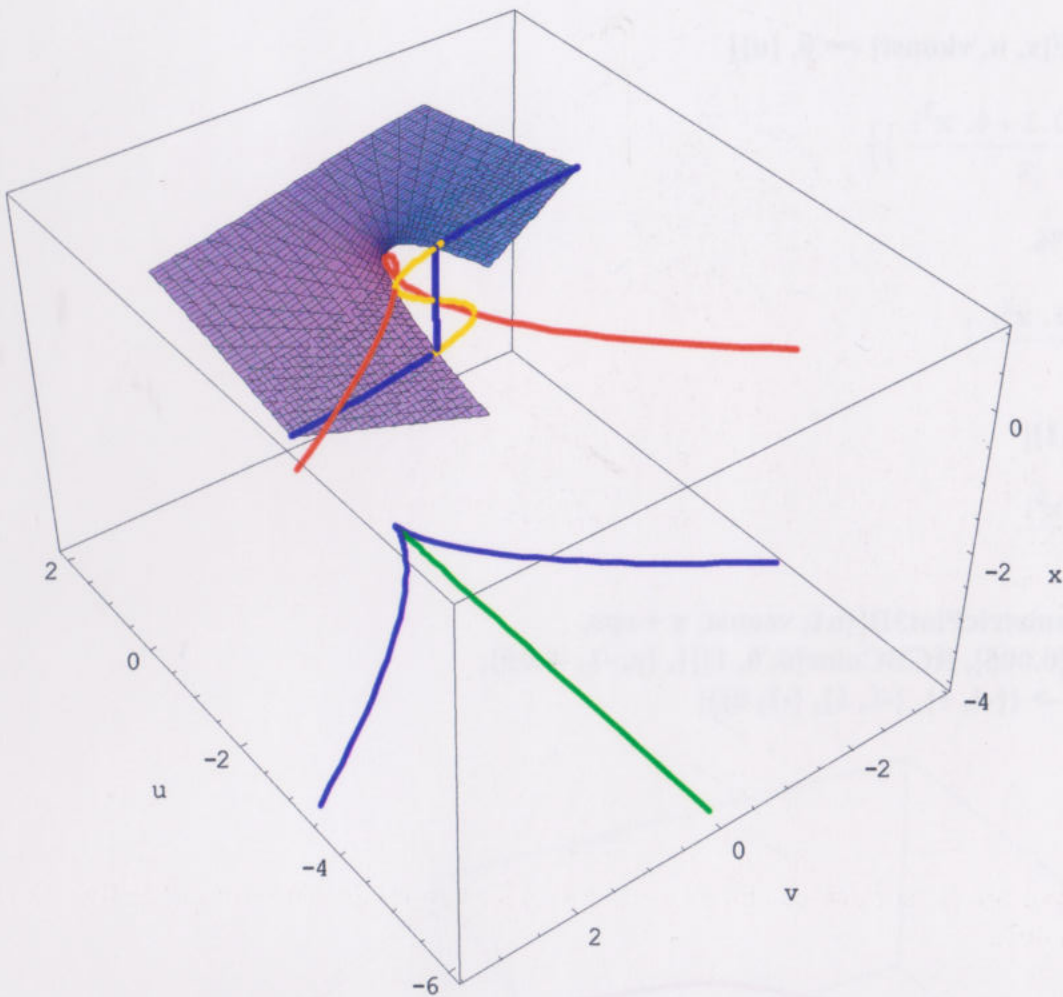
```
sec2u = ParametricPlot3D[{uconst, v2, x + eps,
  {Thickness[0.008], RGBColor[0, 0, 1]}},
  {x, -1, 1}, PlotRange -> {{-5, 5}, {-1, 1.},
  {-2.1, 2.}}, DisplayFunction -> Identity]
```

```
shock = ParametricPlot3D[{s, 0, -4, {Thickness[0.005],
```



```
RGBColor[0, 1, 0]], {x, -1, 1}, DisplayFunction -> Identity]
```

```
Maxwell = Show[C1, C2, C3, sec1u, sec2u, foldline, cusp3D, shock]
```



The curve (line) shown in green color on the display is the Maxwell set of the cusp catastrophe. Above the Maxwell set, F has two minima with the same („critical“) value. Jumps in the system behavior may occur.

Here I have chosen the name „shock“ for the Maxwell set in our picture. This is because there is a connection between the cusp catastrophe and shock waves, that we will meet again in a later chapter.

The last property of the cusp, that we want to demonstrate, is the divergence. Going along the surface (here at constant $v = 0.1$ and -0.1) starting from two points very close to each other, the curves and with them the system show quite different behavior. We do the calculation for the first section:

```
Clear[x, u, vconst, uconst]
eps = 0.1
```

```
0.1
```

vconst = 0.1

0.1

L = Solve[D1F[x, u, vconst] == 0, {u}]

$$\left\{ \left\{ u \rightarrow -\frac{0.5 (0.1 + 4. x^3)}{x} \right\} \right\}$$

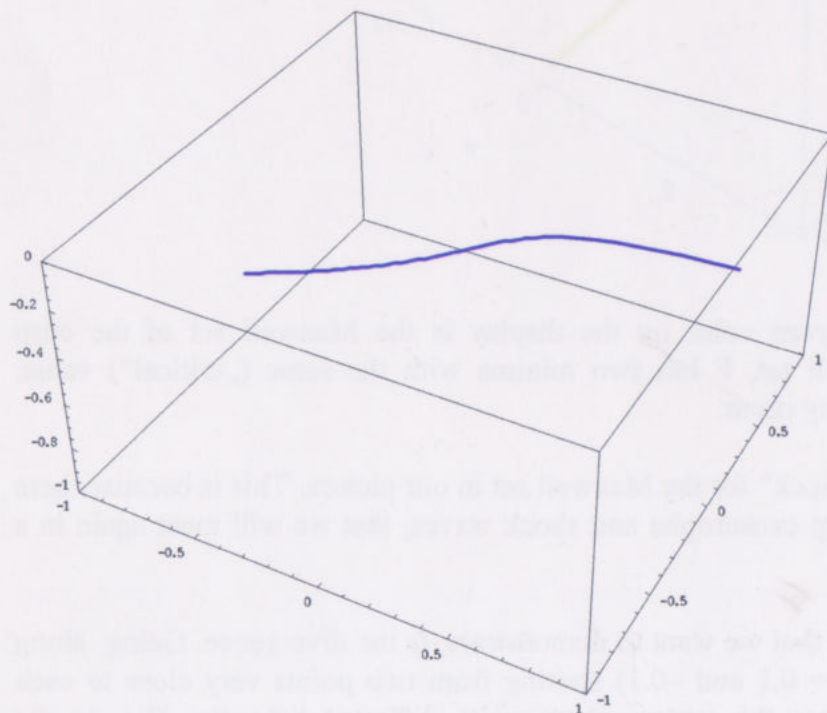
compon = u /. %

$$\left\{ -\frac{0.5 (0.1 + 4. x^3)}{x} \right\}$$

u1 = compon[[1]]

$$-\frac{0.5 (0.1 + 4. x^3)}{x}$$

**diverg1 = ParametricPlot3D[{u1, vconst, x + eps,
{Thickness[0.005], RGBColor[0, 0, 1]}}, {x, -1, -0.05},
PlotRange -> {{-1, 1}, {-1, 1}, {-1, 0}}]**



Now we calculate the same for v=-0.1:

vconst = -0.1

-0.1

L = Solve[D1F[x, u, vconst] == 0, {u}]

$$\left\{ \left\{ u \rightarrow -\frac{0.5(-0.1 + 4. x^3)}{x} \right\} \right\}$$

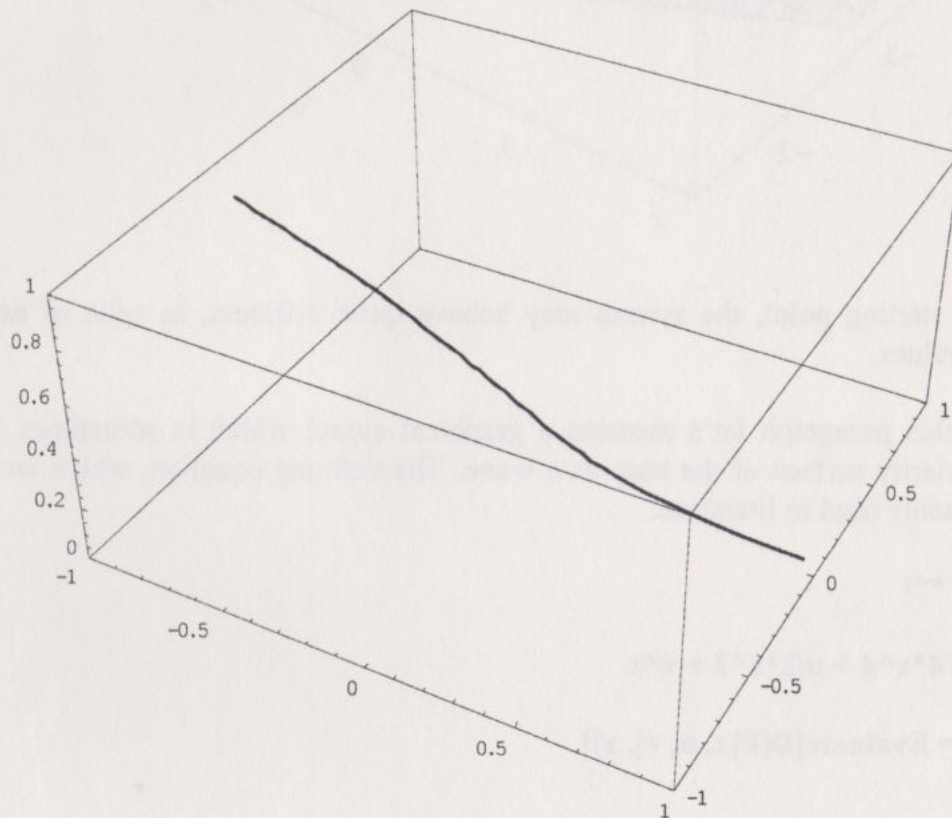
compon = u /. %

$$\left\{ -\frac{0.5(-0.1 + 4. x^3)}{x} \right\}$$

u1 = compon[[1]]

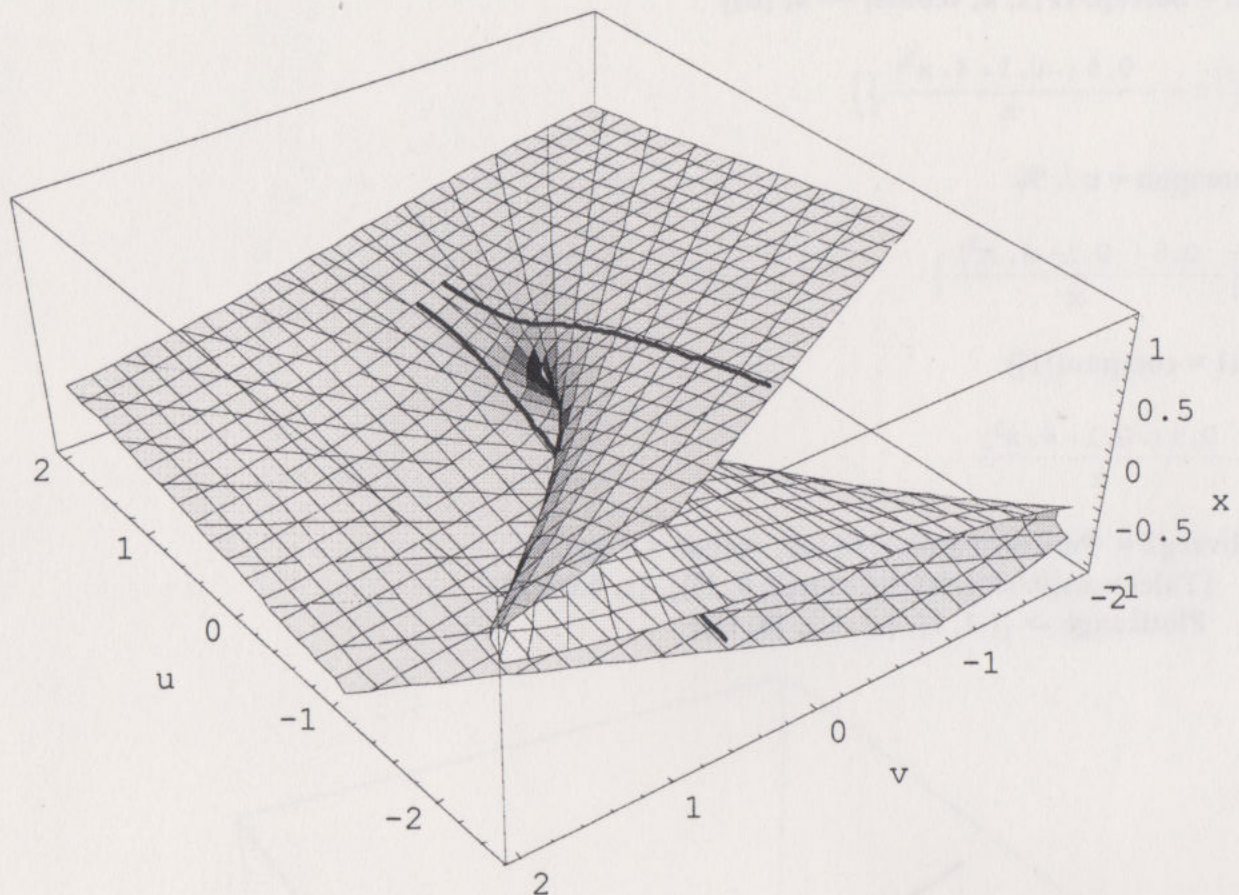
$$-\frac{0.5(-0.1 + 4. x^3)}{x}$$

**diverg2 = ParametricPlot3D[{u1, vconst, x + eps,
{Thickness[0.005], RGBColor[0, 0, 1]}}, {x, 0.05, 1},
PlotRange -> {{-1, 1}, {-1, 1}, {0, 1}}]**



We draw both space curves together with the catastrophe surface:

Show[CriticalPoints, diverg1, diverg2, ColorOutput->CMYKColor]



According to the starting point, the system may behave quite different, in spite of nearly identical starting values.

At the end of this paragraph let's mention a graphical aspect which is sometimes very helpful: The singularity surface of the cusp as a wave. The defining equation, which we use here, is also commonly used in literature.

```
Remove["Global`*"]
```

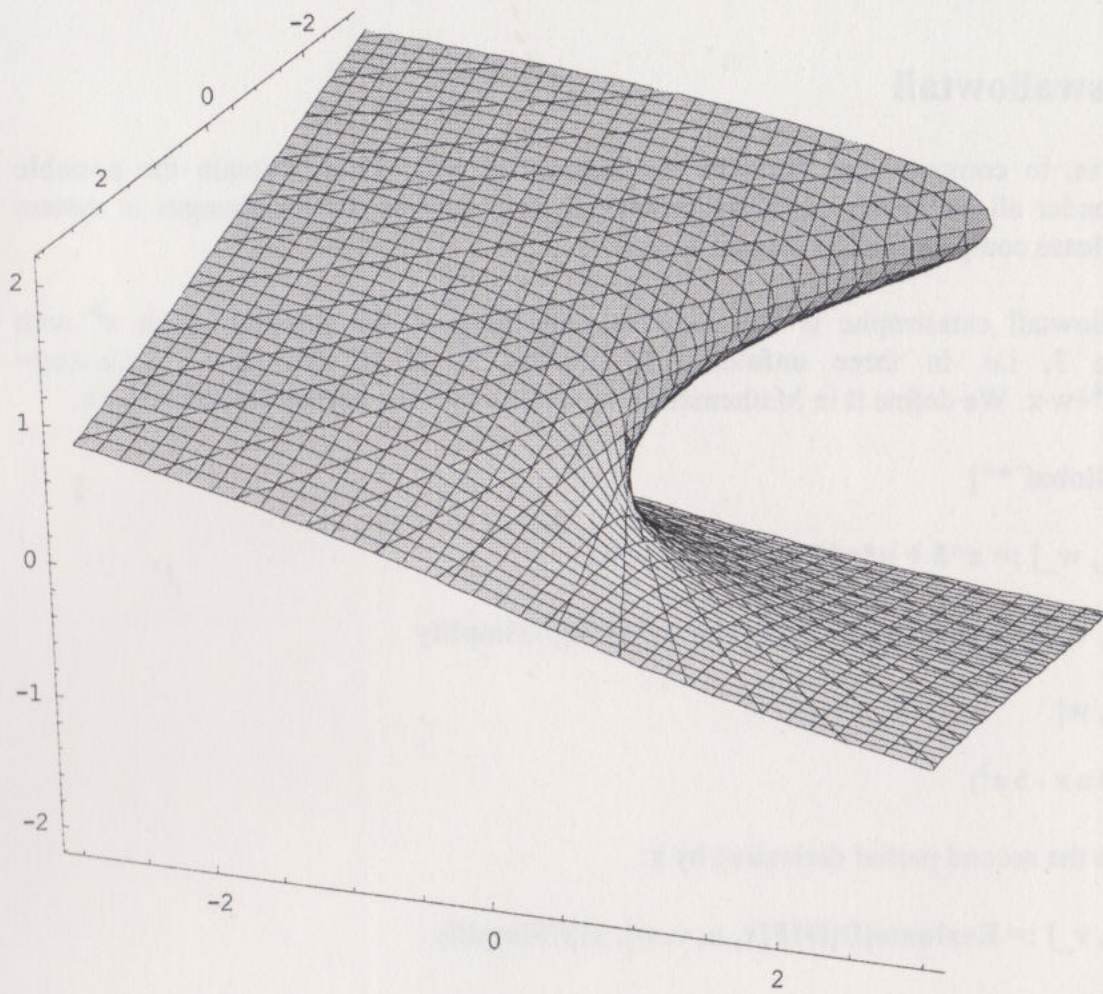
```
F[x_, u_, v_] := 1/4*x^4 + u/2*x^2 + v*x
```

```
D1F[x_, u_, v_] := Evaluate[D[F[x, u, v], x]]
```

```
<< "Graphics`Master`"
```

```
CriticalPoints =
```

```
ContourPlot3D[D1F[x, u, v], {u, -3, 3}, {v, -3, 3}, {x, -2.5, 2.5},  
PlotPoints -> 7, ViewPoint -> {5, 1.5, 2}, Axes -> True,  
Boxed -> False]
```

7.3 The swallowtail

Our aim is, to compute and visualize the bifurcation sets. These contain the possible candidates under all parameter constellations which may lead to sudden changes in system behavior. (Please compare the last section above).

The swallowtail catastrophe is the universal unfolding of the function germ x^5 with codimension 3, i.e. in three unfolding parameters. It is of the form $F(x,u,v,w)=x^5+u\cdot x^3+v\cdot x^2+w\cdot x$. We define it in Mathematica, as well as the first partial derivative by x .

```
Remove["Global`*"]
```

```
F[x_, u_, v_, w_] := x^5 + u*x^3 + v*x^2 + w*x
```

```
D1F[x_, u_, v_, w_] := Evaluate[D[F[x, u, v, w], x]]//Simplify
```

```
D1F[x, u, v, w]
```

```
w + x (2 v + 3 u x + 5 x^3)
```

We compute the second partial derivative by x :

```
D2F[x_, u_, v_] := Evaluate[D[D1F[x, u, v, w], x]]//Simplify
```

```
D2F[x, u, v]
```

```
2 (v + 3 u x + 10 x^3)
```

The singularity surface (catastrophe manifold) can not completely be drawn in one picture, as we have done it with the cusp, since here the parameter space is already 3-dimensional and the direction of the state variable x is one more dimension. So we first have to make sections, for example for constant values of u . Assume $u=-1$. For this value we determine again the first and second partial derivative

```
u = -1;
```

```
D1uconstF[x_, v_, w_] := D1F[x, u, v, w]
```

```
D1uconstF[x, v, w]
```

```
w + x (2 v - 3 x + 5 x^3)
```

```
D2uconstF[x_, v_] := D2F[x, u, v]
```

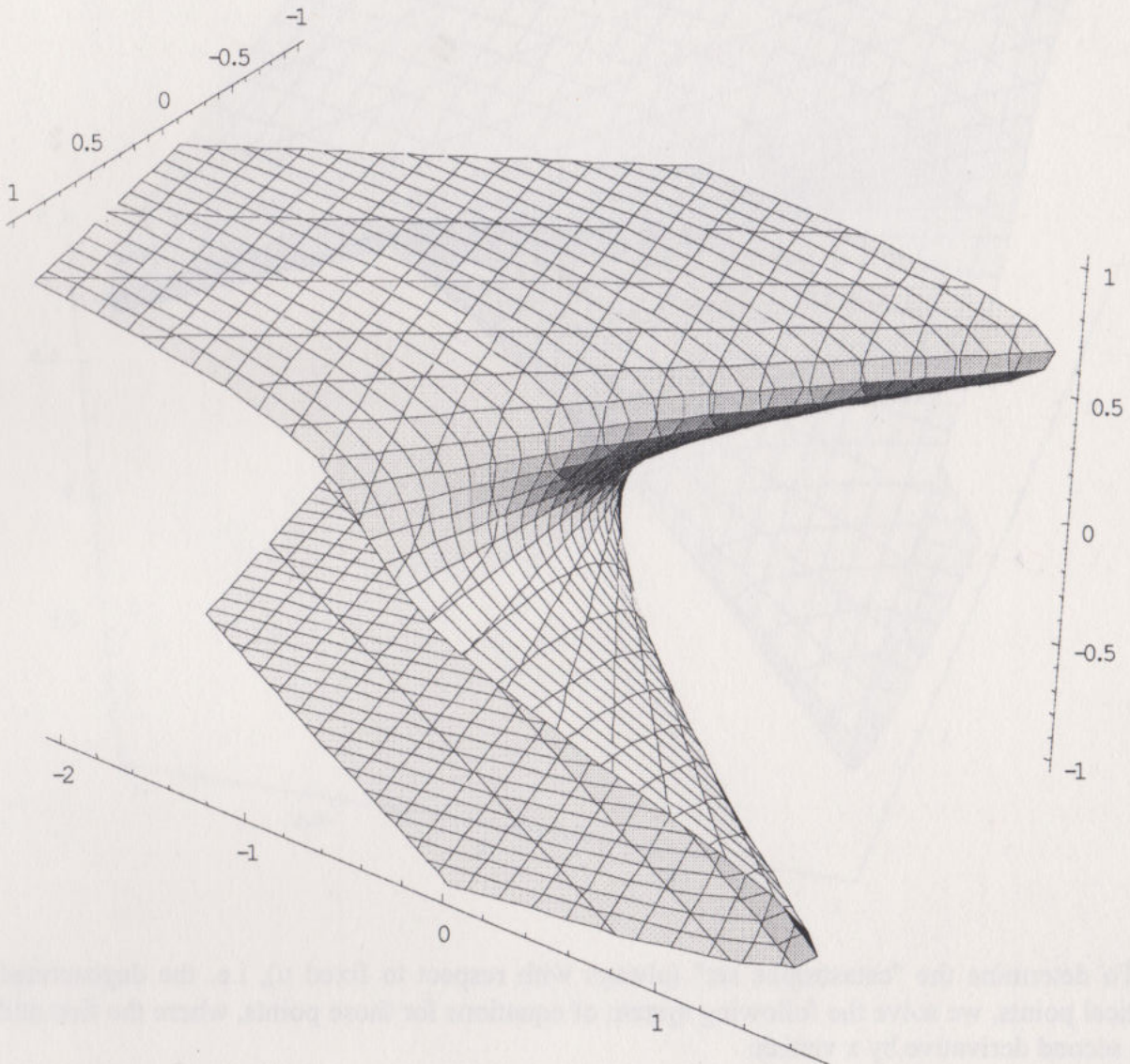
```
D2uconstF[x, v]
```

```
2 (v - 3 x + 10 x^3)
```


We draw for example the critical points belonging to the value $u=-1$:

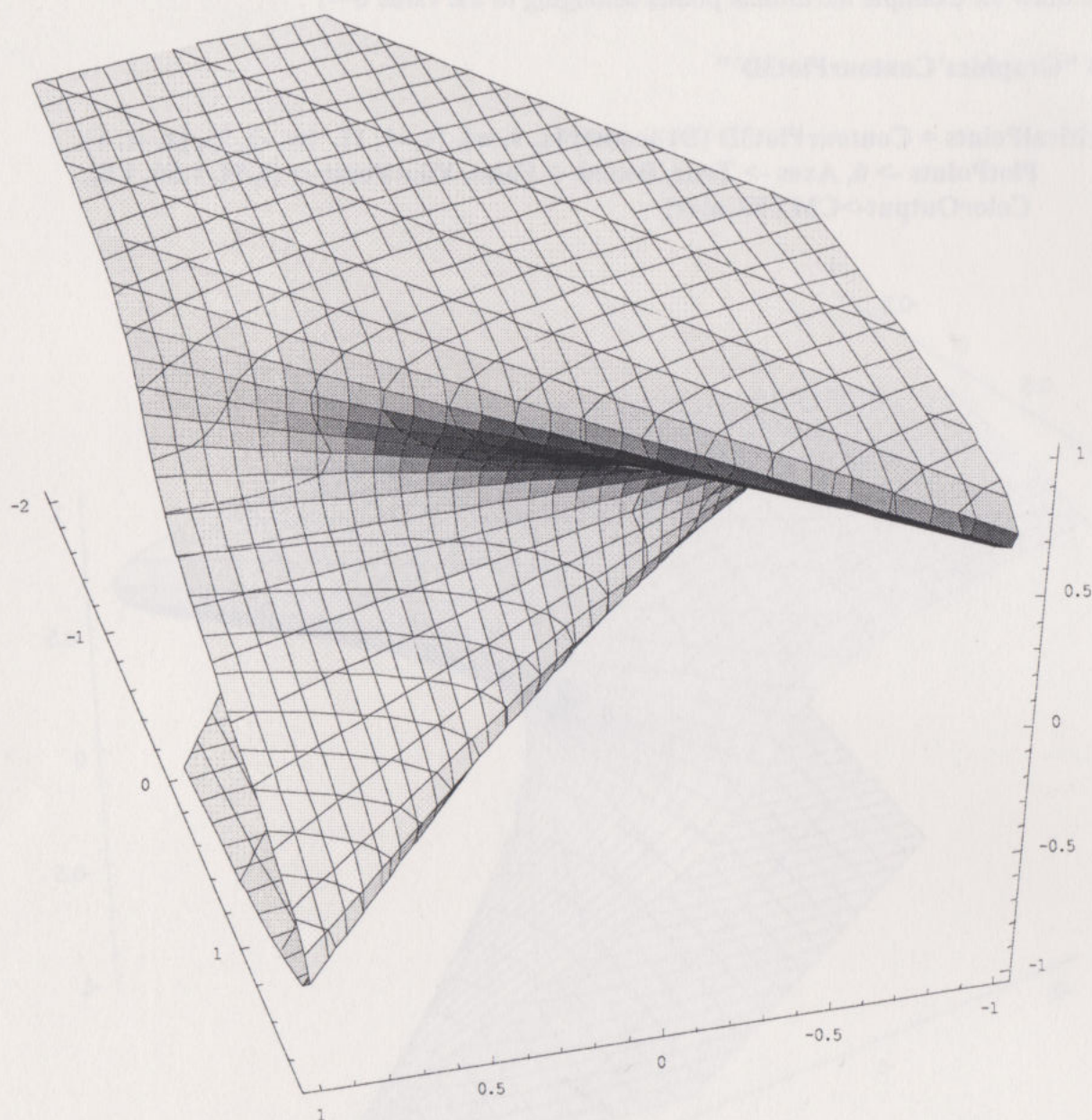
```
<< "Graphics`ContourPlot3D`"
```

```
CriticalPoints = ContourPlot3D [D1uconstF[x, v, w], {v, -1, 1}, {w, -2, 2}, {x, -1, 1},  
PlotPoints -> 6, Axes -> True, Boxed -> False, ViewPoint -> {2.54, 1.56, 1.6},  
ColorOutput->CMYKColor]
```



From another viewpoint:

```
CPov = Show[CriticalPoints, ViewPoint -> {0.77, 2.73, 1.717}]
```

To determine the "catastrophe set" (always with respect to fixed u), i.e. the degenerated critical points, we solve the following system of equations for those points, where the first and the second derivative by x vanish:

```
Solu1 = Solve[{D1uconstF[x, v, w] == 0, D2uconstF[x, v] == 0}, {v, w}]/Flatten
```

```
{ {w → 3 (-x2 + 5 x4) , v → 3 x - 10 x3 } }
```

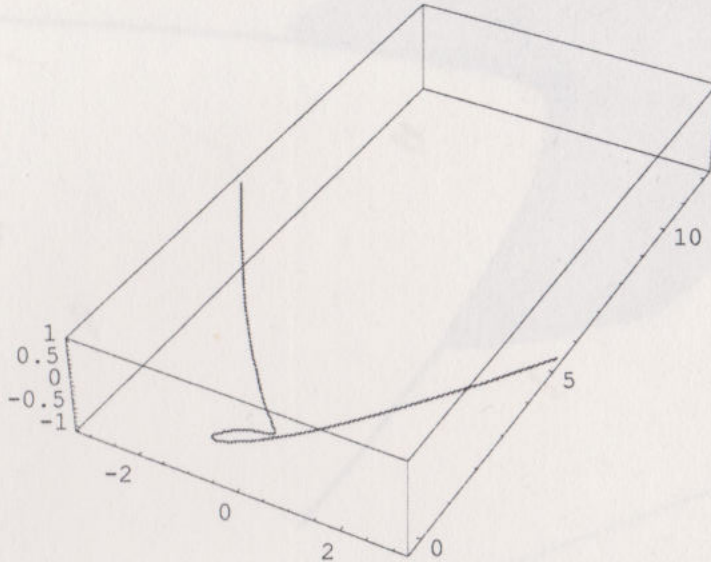
The solutions for v and w are called s and t .

```
{s, t} = Simplify[{v, w} /. Solu1]
```

```
{ 3 x - 10 x3, 3 x2 (-1 + 5 x2) }
```


We draw the catastrophe set ("fold line"). Hereto we use the option "ParametricPlot3D", at the second time we use it together with the option "Viewpoint", in order to show the space curve from another perspective.

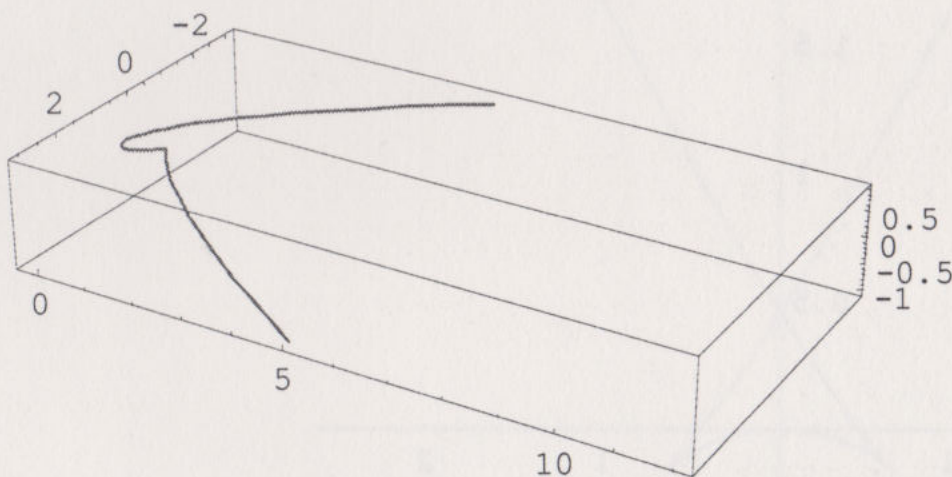
```
fold = ParametricPlot3D[{s, t, x,
  {Thickness[0.005], RGBColor[1, 0, 0]}}, {x, -1, 1}]
```



The following constant $\text{eps}=0.05$ is used for the reason of better visibility in the composed picture.

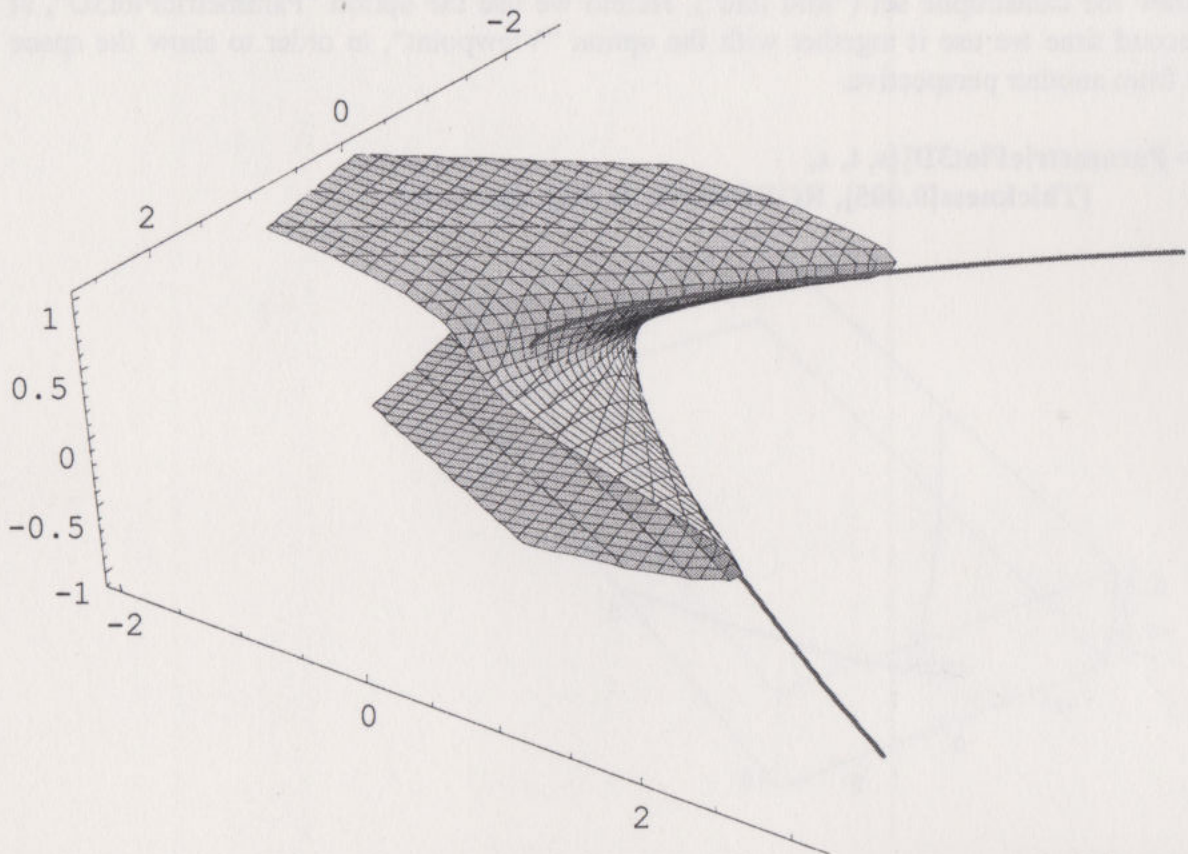
```
eps = -0.05;
```

```
FLov = ParametricPlot3D[{s, t, x + eps,
  {Thickness[0.005], RGBColor[1, 0, 0]}}, {x, -1, 1},
  ViewPoint -> {2.5, 1.5, 1.5}]
```



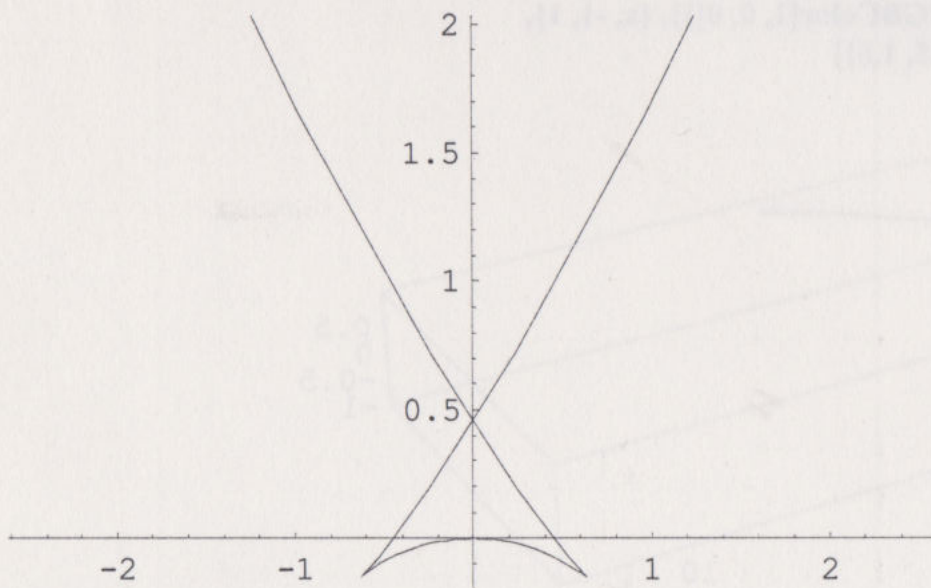
Now we draw the composed picture:

```
Piccompose1 = Show[CriticalPoints, FLov]
```



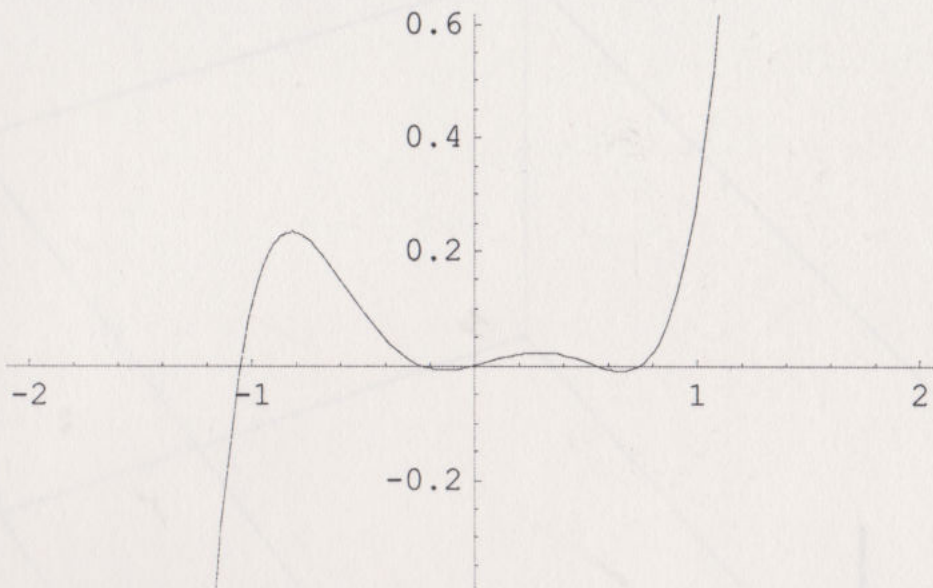
The projection of the fold line onto the “restricted” parameter space (v - w space) gives the typical form of a swallowtail:

InPara = ParametricPlot[{s, t}, {x, -1, 1}]



Just as with the cusp, you may draw the forms of the Function F , which belong to different regions of v - w parameter space ($u=-1$), and which are defined by the swallowtail curve above. An example shows F depending on x at the parameter constellation $u=-1, v=0.2$ and $w=0.1$

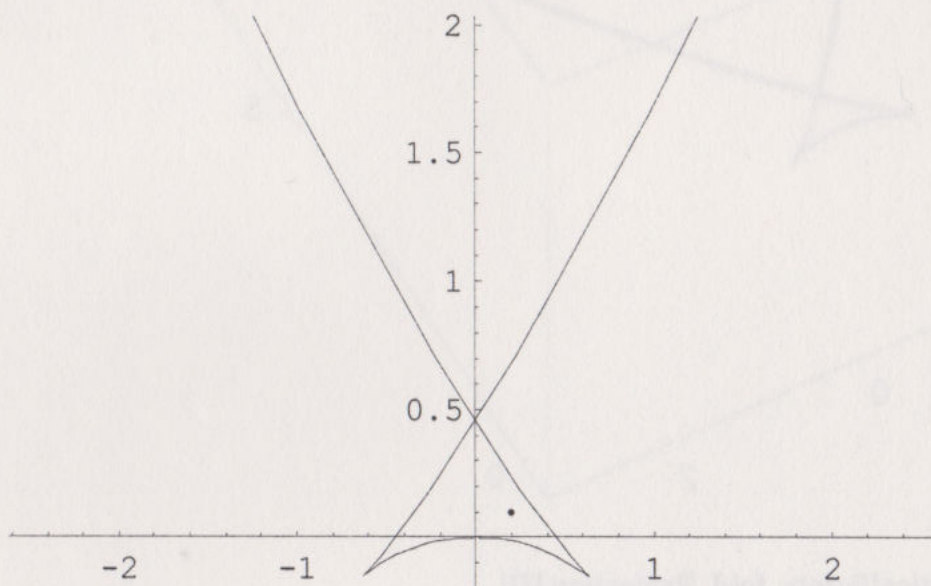
Plot[F[x, -1, 0.2, 0.1], {x, -2, 2}]



We show the position of this special parameter values as point in v-w space ($u=-1$).

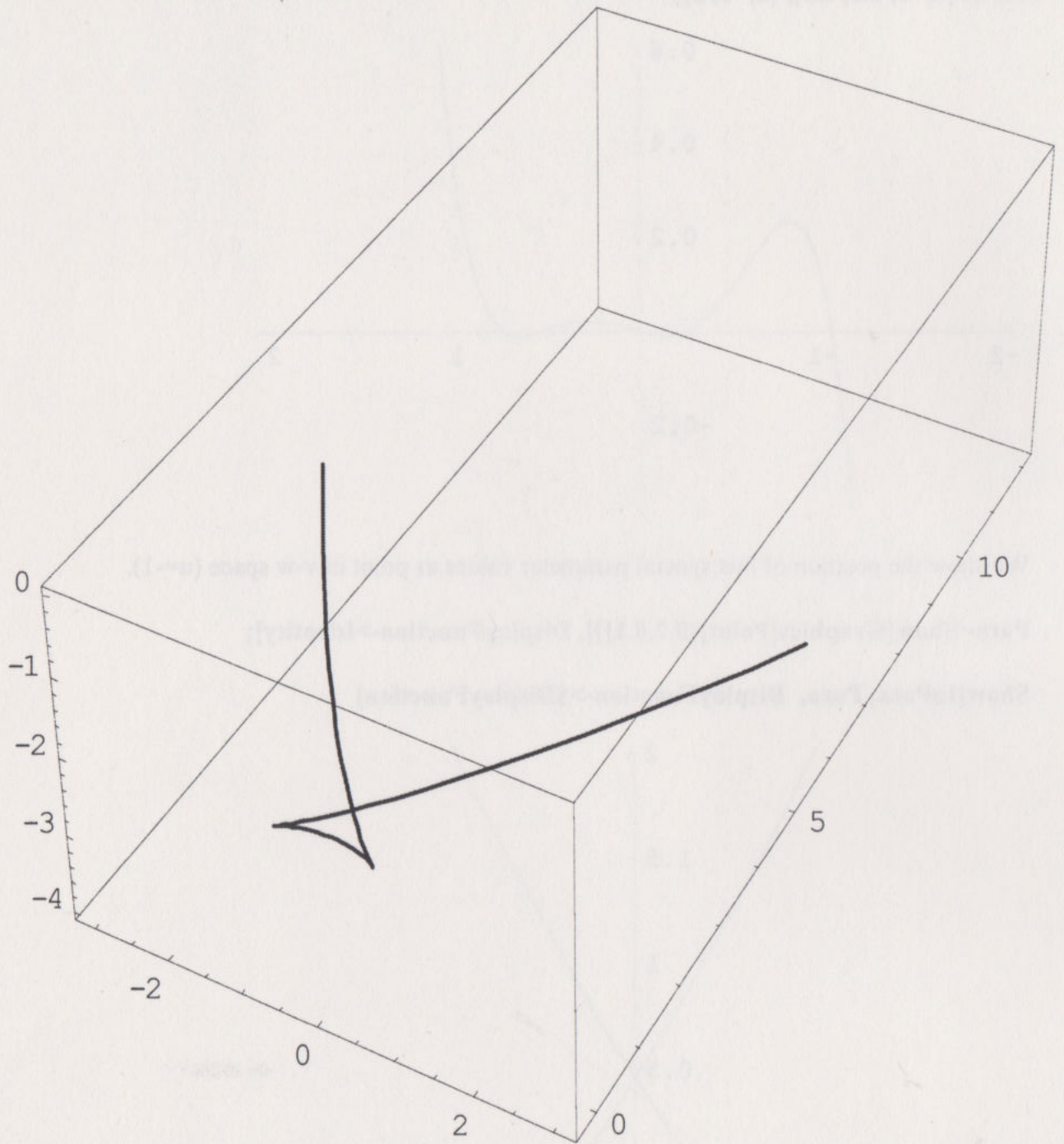
Para=Show[Graphics[Point[{0.2,0.1}]], DisplayFunction->Identity];

Show[InPara, Para, DisplayFunction->\$DisplayFunction]

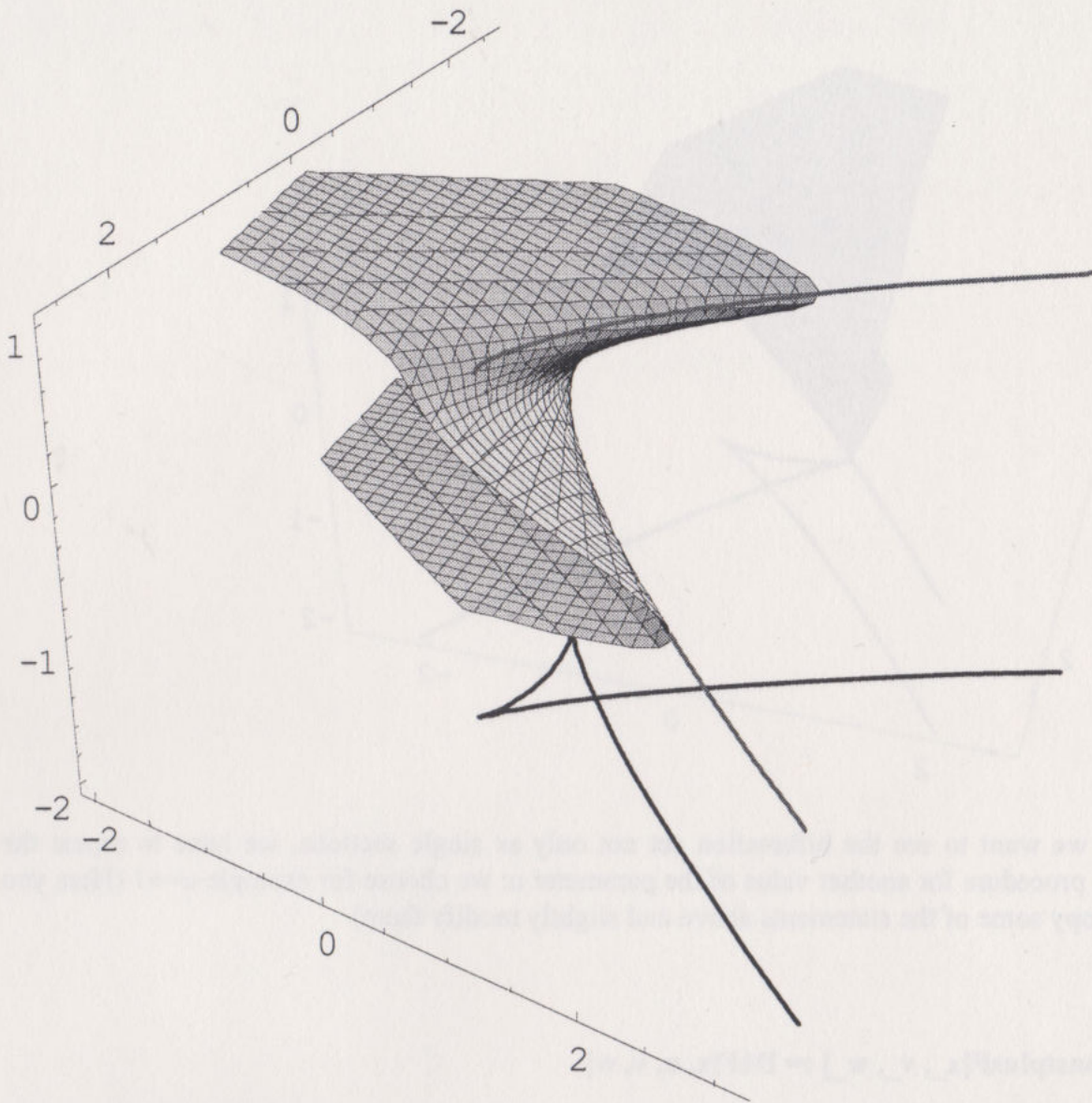


Now we want to show in a single graphic the projection of the fold line together with the singularity surface (here always for the constant parameter value $u=-1$).

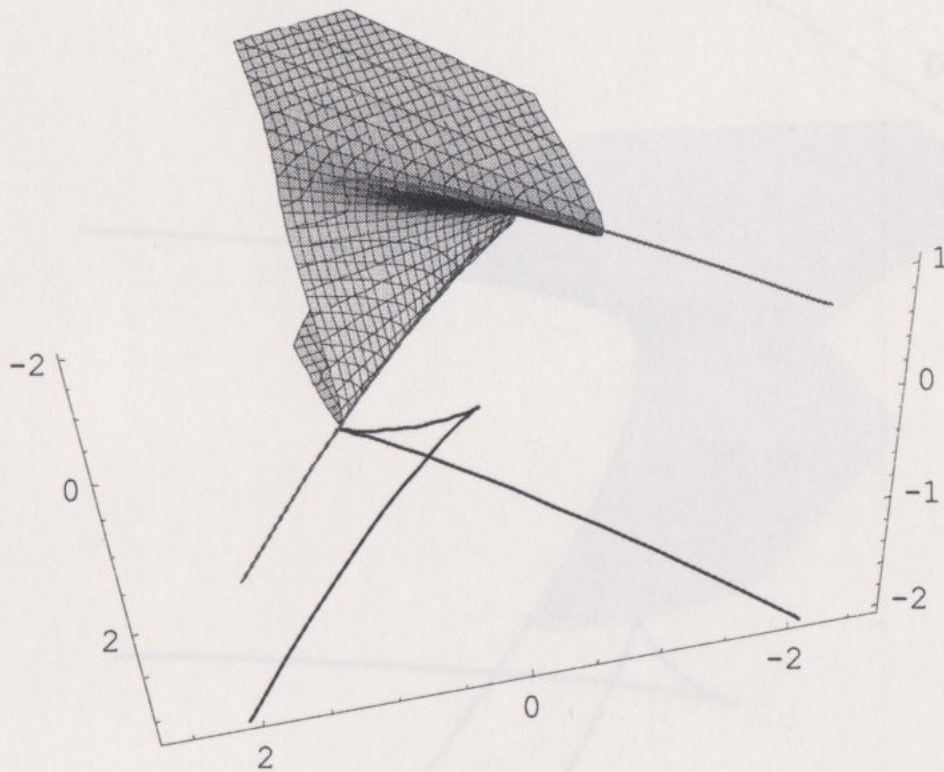
**Projection3D = ParametricPlot3D[{s, t, -2, {Thickness[0.005],
RGBColor[0, 0, 1]}}, {x, -1, 1}]**



Piccomp2 = Show[CriticalPoints, fold, Projection3D]



Picov = Show[CPov, fold,
Projection3D]



Since we want to see the bifurcation set not only as single sections, we have to repeat the whole procedure for another value of the parameter u : we choose for example $u=+1$ (Here you may copy some of the statements above and slightly modify them)

$u = 1;$

$D1uconstplusF[x_, v_, w_] := D1F[x, u, v, w]$

$D1uconstplusF[x, v, w]$

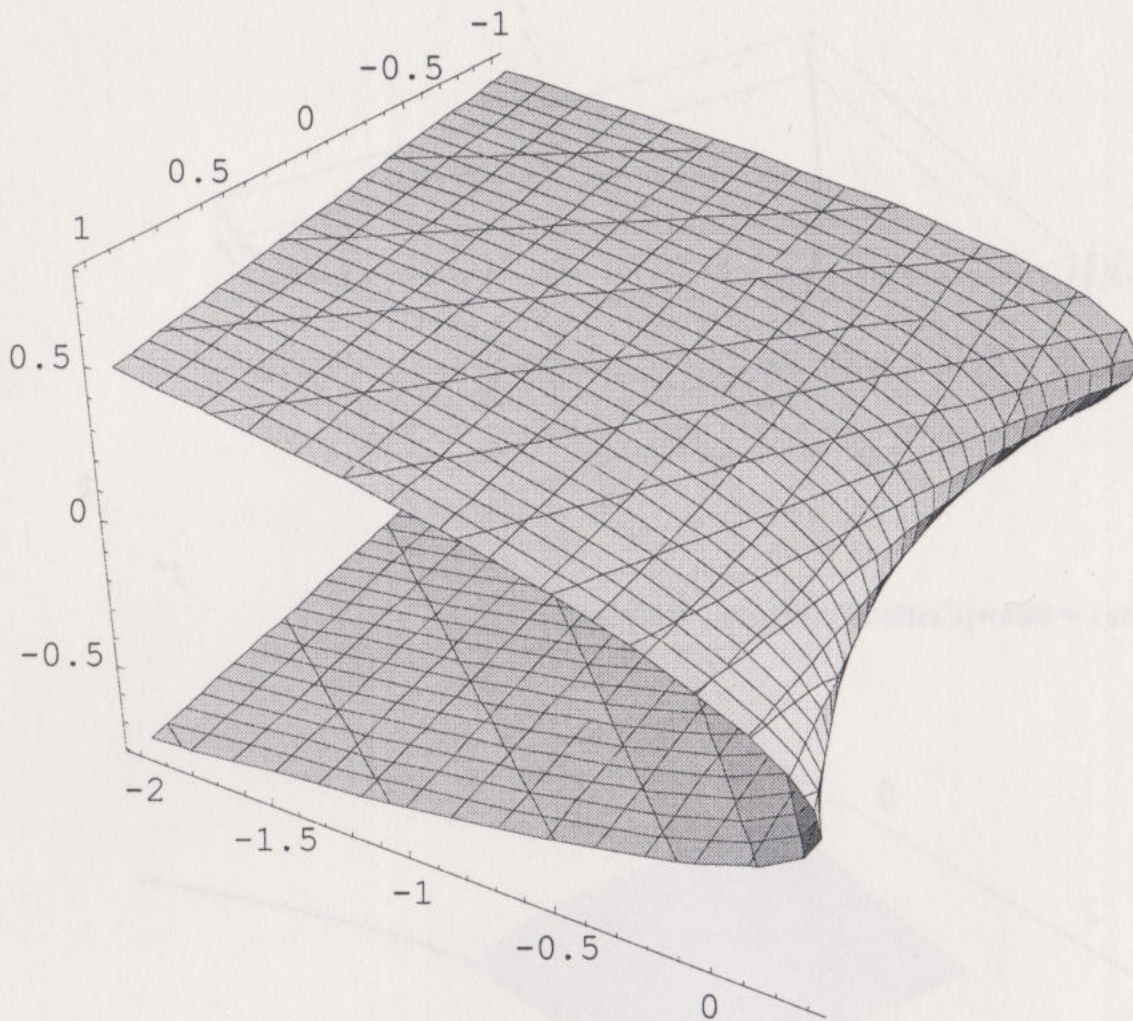
$w + x (2 v + 3 x + 5 x^3)$

$D2uconstplusF[x_, v_] := D2F[x, u, v]$

$D2uconstplusF[x, v]$

$2 (v + 3 x + 10 x^3)$

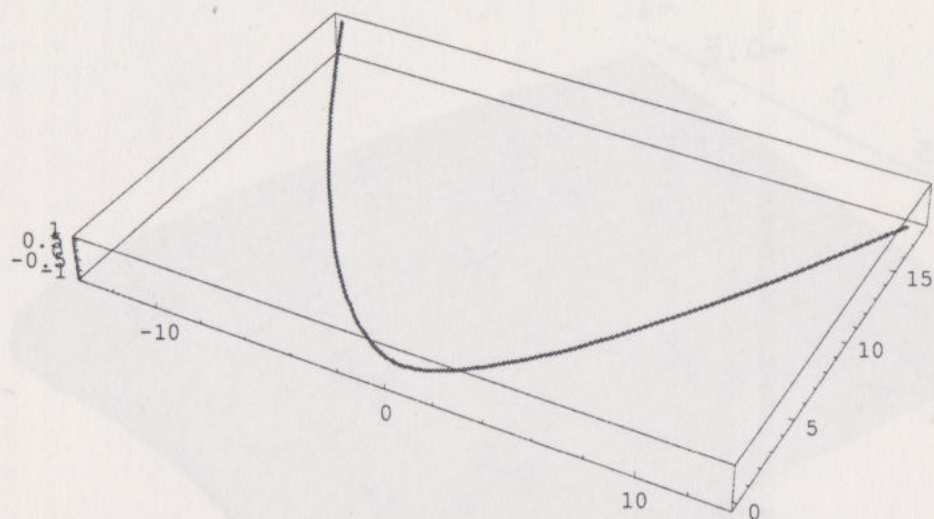
**$CriticalPointsplus = ContourPlot3D[D1uconstF[x, v, w], \{v, -1, 1\},$
 $\{w, -2, 2\}, \{x, -1, 1\}, PlotPoints \rightarrow 6, Axes \rightarrow True,$
 $Boxed \rightarrow False, ViewPoint \rightarrow \{2.54, 1.56, 1.6\}]$**



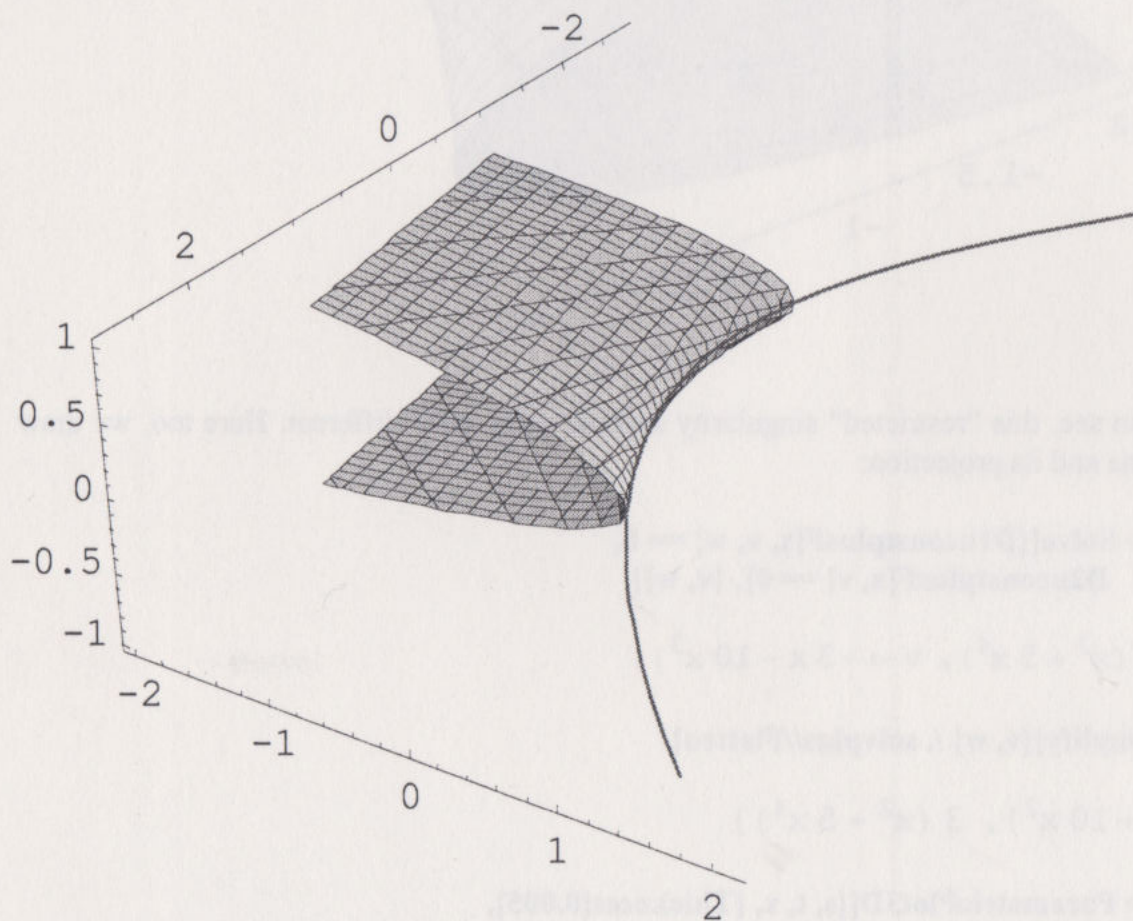
As you can see, this “restricted” singularity surface looks quite different. Here too, we draw the fold line and its projection:

$$\text{solvplus} = \text{Solve}[\{\text{D1uconstplusF}[x, v, w] == 0, \text{D2uconstplusF}[x, v] == 0\}, \{v, w\}]$$
$$\{\{w \rightarrow 3(x^2 + 5x^4), v \rightarrow -3x - 10x^3\}\}$$
$$\{s, t\} = \text{Simplify}[\{v, w\} /. \text{solvplus} // \text{Flatten}]$$
$$\{-x(3 + 10x^2), 3(x^2 + 5x^4)\}$$

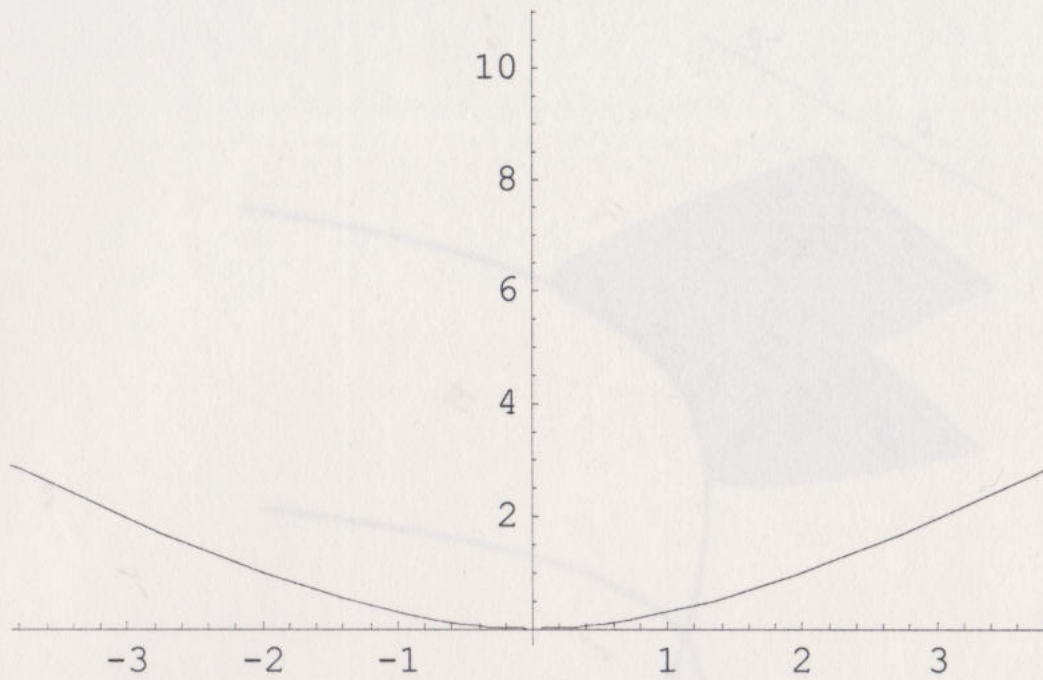
```
foldplus = ParametricPlot3D[{s, t, x, {Thickness[0.005],  
RGBColor[1, 0, 0]}}, {x, -1, 1}];
```



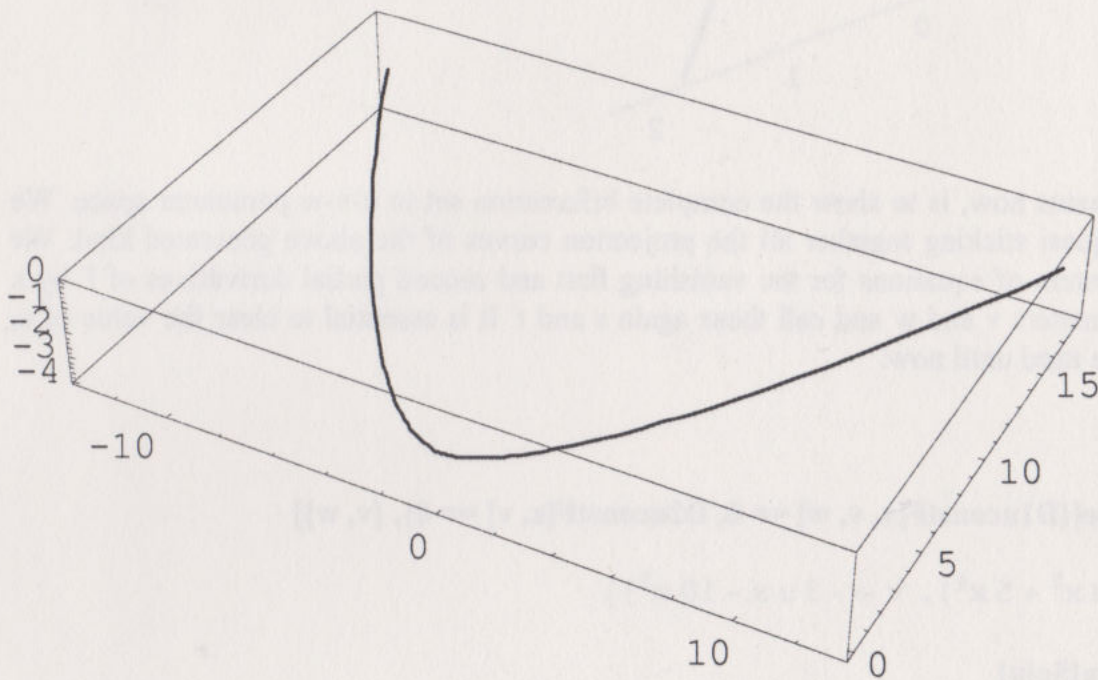
Piccompose1 = Show[CriticalPointsplus, foldplus];



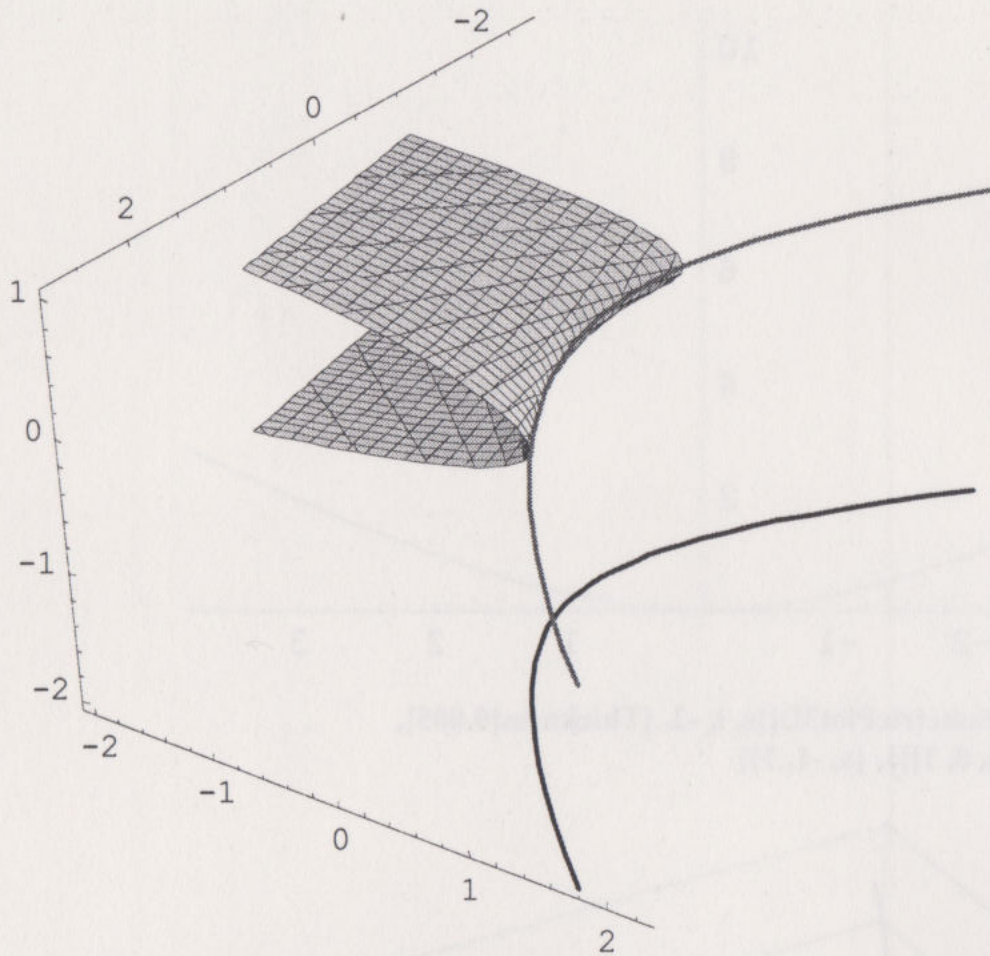
Inparaplaus = ParametricPlot[{s, t}, {x, -1, 1}]



```
Projection3D = ParametricPlot3D[{s, t, -2, {Thickness[0.005],
  RGBColor[0, 0, 1]}}], {x, -1, 1}]
```



```
Piccomp3 = Show[CriticalPointsplus, foldplus, Projection3D]
```



What remains now, is to show the complete bifurcation set in u - v - w parameter space. We do this by quasi sticking together all the projection curves of the above generated kind. We solve the system of equations for the vanishing first and second partial derivatives of F by x for the parameters v and w and call these again s and t . It is essential to clear the value of u , that we have used until now:

```
Clear[u]
```

```
Solu = Solve[{D1uconstF[x, v, w] == 0, D2uconstF[x, v] == 0}, {v, w}]
```

```
{{w -> 3 (u x^2 + 5 x^4), v -> -3 u x - 10 x^3}}
```

```
L1 = Flatten[Solu]
```

```
{w -> 3 (u x^2 + 5 x^4), v -> -3 u x - 10 x^3}
```

```
Clear[s, t]
```

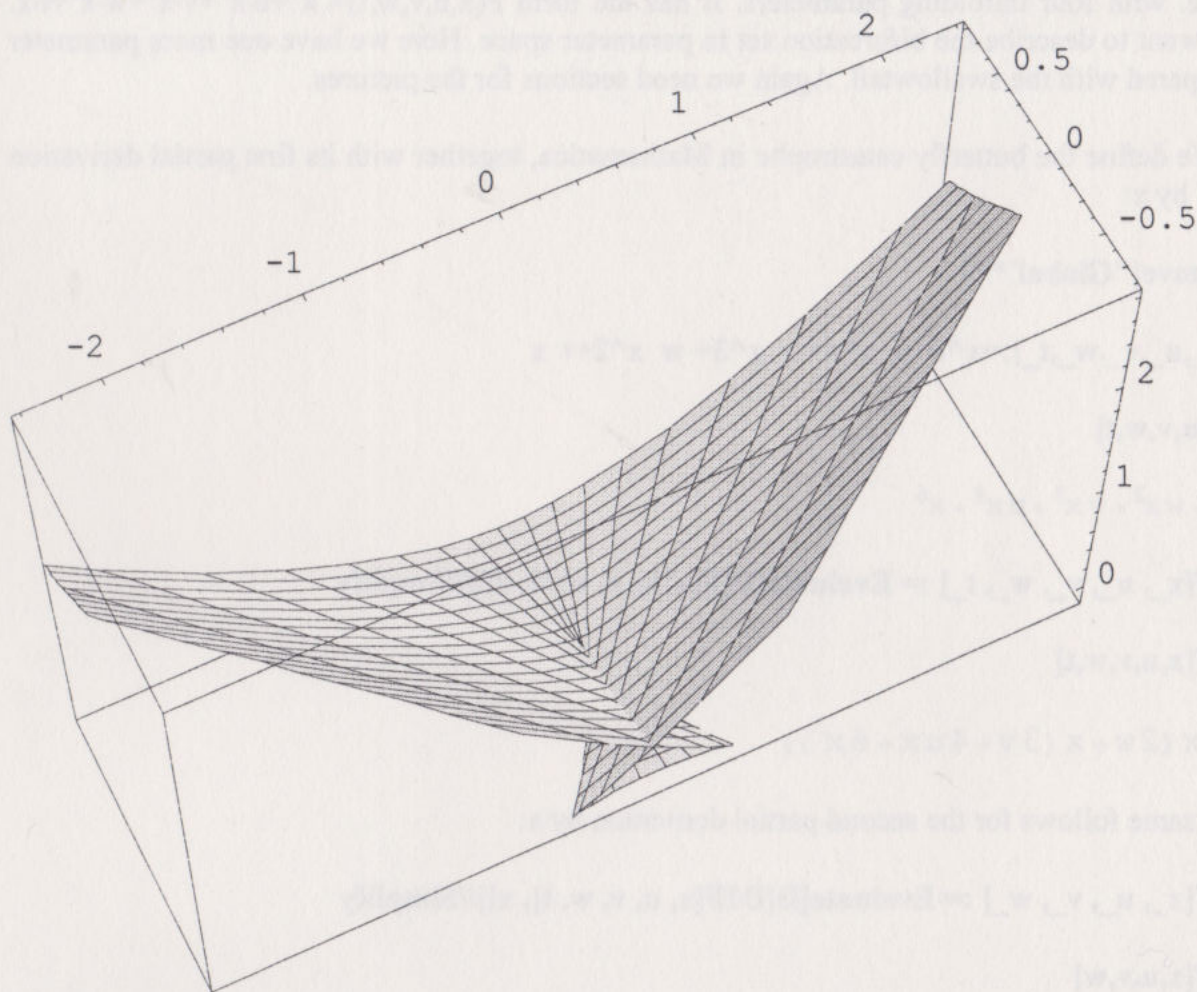
```
{s, t} = Simplify[{v, w] /. L1}
```

```
{-3 u x - 10 x^3, 3 x^2 (u + 5 x^2)}
```

Now we get the desired result with the help of the statement "ParametricPlot3D". The variable

u takes positive values as well as negative values. v and w are parameterized here by x .

```
Fl3Dplus = ParametricPlot3D[{s, u, t}, {x, -0.6, 0.6}, {u, -0.8, 0.8},  
ViewPoint -> {-1, -2, 4.}, ColorOutput -> CMYKColor]
```



7.4 The butterfly

The butterfly catastrophe is the universal unfolding of the function germ x^6 of codimension 4, i.e. with four unfolding parameters. It has the form $F(x,u,v,w,t) = x^6 + u \cdot x^4 + v \cdot x^3 + w \cdot x^2 + t \cdot x$. We want to describe the bifurcation set in parameter space. Here we have one more parameter compared with the swallowtail. Again we need sections for the pictures.

We define the butterfly catastrophe in Mathematica, together with its first partial derivation D1F by x:

```
Remove["Global`*"]
```

```
F[x_,u_,v_,w_,t_]:=x^6+u x^4+ v x^3+ w x^2+t x
```

```
F[x,u,v,w,t]
```

$$t x + w x^2 + v x^3 + u x^4 + x^6$$

```
D1F[x_, u_, v_, w_, t_] := Evaluate[D[F[x, u, v, w, t], x]]//Simplify
```

```
D1F[x,u,v,w,t]
```

$$t + x (2 w + x (3 v + 4 u x + 6 x^3))$$

The same follows for the second partial derivation by x:

```
D2F[x_, u_, v_, w_] := Evaluate[D[D1F[x, u, v, w, t], x]]//Simplify
```

```
D2F[x,u,v,w]
```

$$2 (w + 3 x (v + 2 u x + 5 x^3))$$

We fix two parameters arbitrarily, e.g. $u=-1$ and $v=0$. The first and second partial derivations for these constant values get new names:

```
u=-1;
```

```
v=0;
```

```
D1uvconstF[x_,w_,t_]:=D1F[x,u,v,w,t]
```

```
D1uvconstF[x,w,t]
```

$$t + 2 w x - 4 x^3 + 6 x^5$$

```
D2uvconstF[x_,w_]:=D2F[x,u,v,w]
```


D2uvconstF[x,w]

$$2 (w - 6 x^2 + 15 x^4)$$

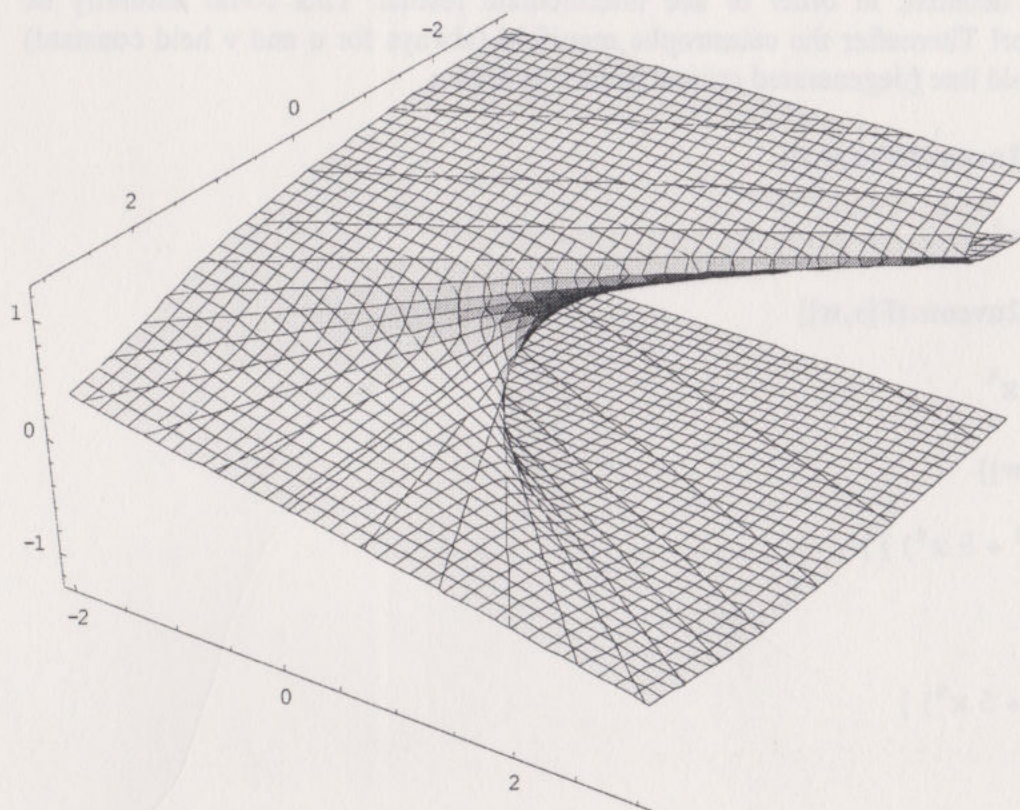
We load the package "Graphics`Contourplot3D":

<< "Graphics`ContourPlot3D"

Then we draw the set of the critical points for $u=-1$ and $v=0$.

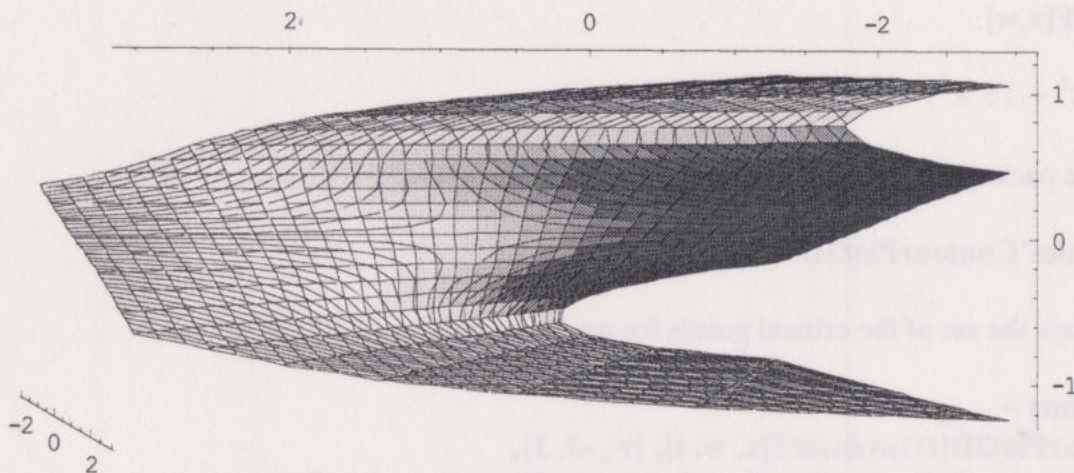
CriticalPoints =

```
ContourPlot3D[D1uvconstF[x, w, t], {w, -3, 3},
{t, -2, 3}, {x, -2, 2}, PlotPoints -> 7, Axes -> True, Boxed -> False,
ViewPoint -> {2.5, 1.56, 1.6}]
```



From another viewpoint:

```
CPov = Show[CriticalPoints, ViewPoint -> {1, 4, 0.1}, ColorOutput->CMYKColor]
```



The next input lines have the aim to draw the fold line of this surface. Some statements are programmed very detailed, in order to see intermediate results. This could naturally be programmed shorter! Thereafter the catastrophe manifold (always for u and v held constant) together with the fold line (degenerated critical points) is drawn.

```
testd1=Expand[D1uvconstF[x,w,t]]
```

$$t + 2wx - 4x^3 + 6x^5$$

```
testd2=Expand[D2uvconstF[x,w]]
```

$$2w - 12x^2 + 30x^4$$

```
Solve[testd2==0,{w}]
```

$$\{ \{w \rightarrow -3(-2x^2 + 5x^4)\} \}$$

```
Flatten[%]
```

$$\{w \rightarrow -3(-2x^2 + 5x^4)\}$$

```
wtest=w/.%
```

$$-3(-2x^2 + 5x^4)$$

```
testd1 /. w->wtest
```

$$t - 4x^3 + 6x^5 - 6x(-2x^2 + 5x^4)$$

```
Simplify[%]
```

$$t + 8x^3 - 24x^5$$

```
Solve[%==0,{t}]
```

$$\{t \rightarrow 8(-x^3 + 3x^5)\}$$

Flatten[%]

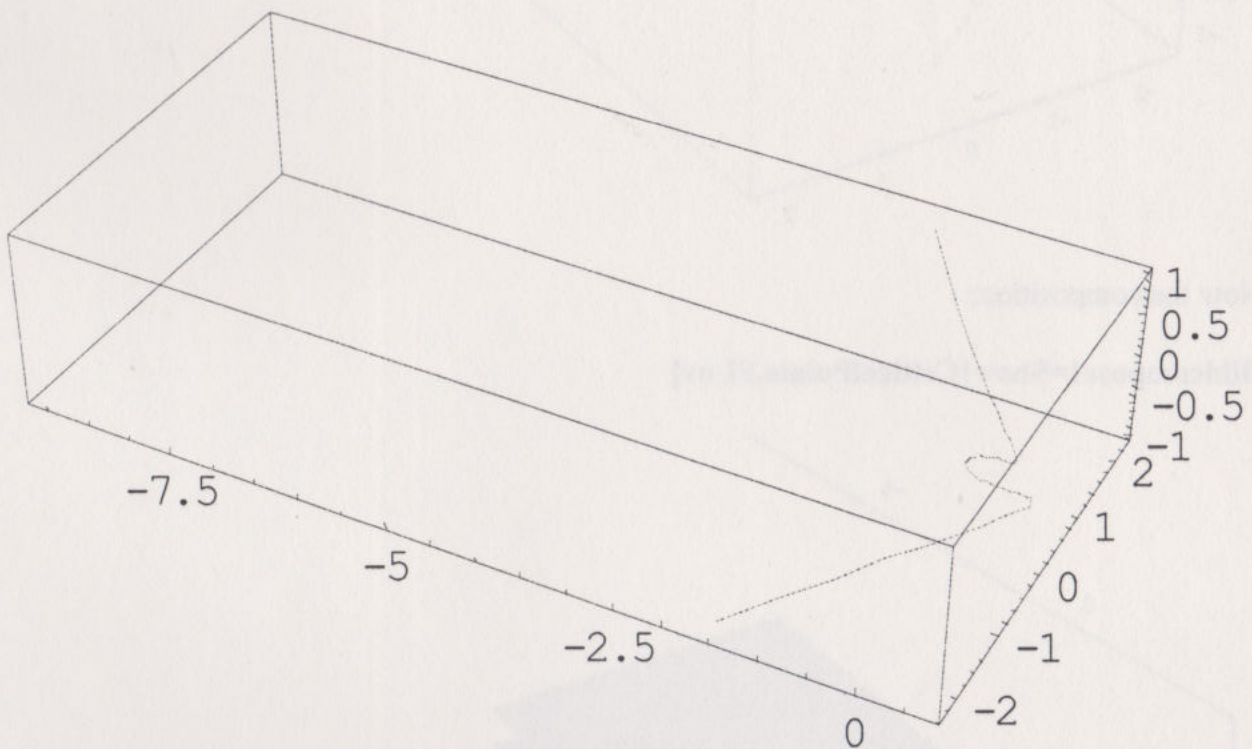
```
{t → 8 (-x3 + 3 x5) }
```

ttest=t./%

```
8 (-x3 + 3 x5)
```

fold=

```
ParametricPlot3D[{wtest,ttest,x,{Thickness[0.002],  
  RGBColor[1, 0, 0]}},{x,-1, 1}]
```

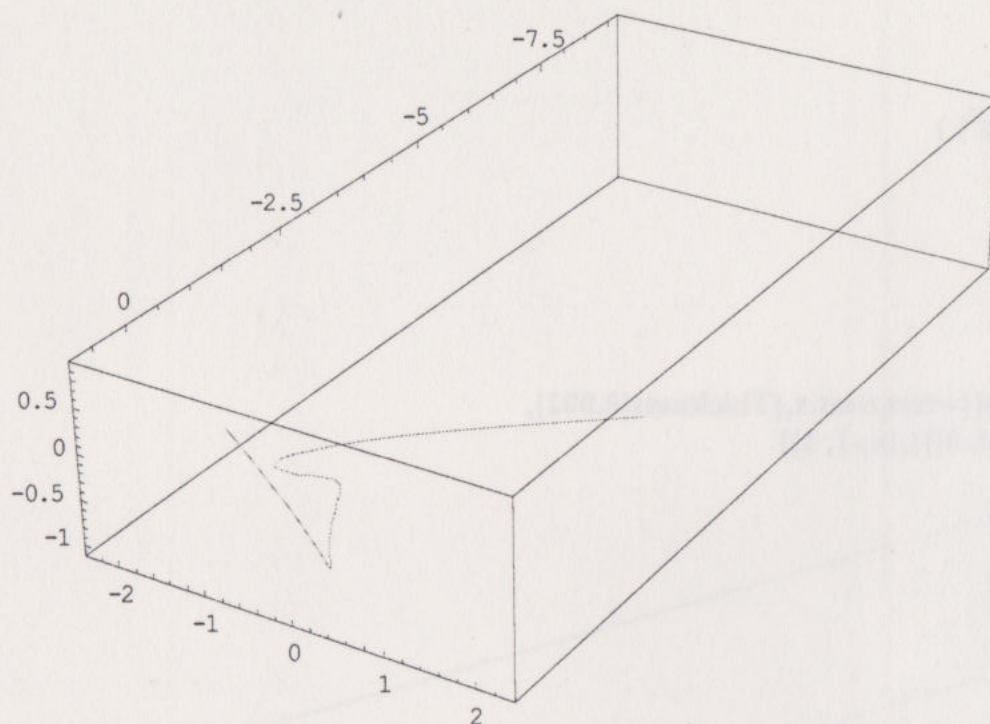


eps again is for the reason of better visibility of the curve in the next composition of the graphics.

eps=-0.05;

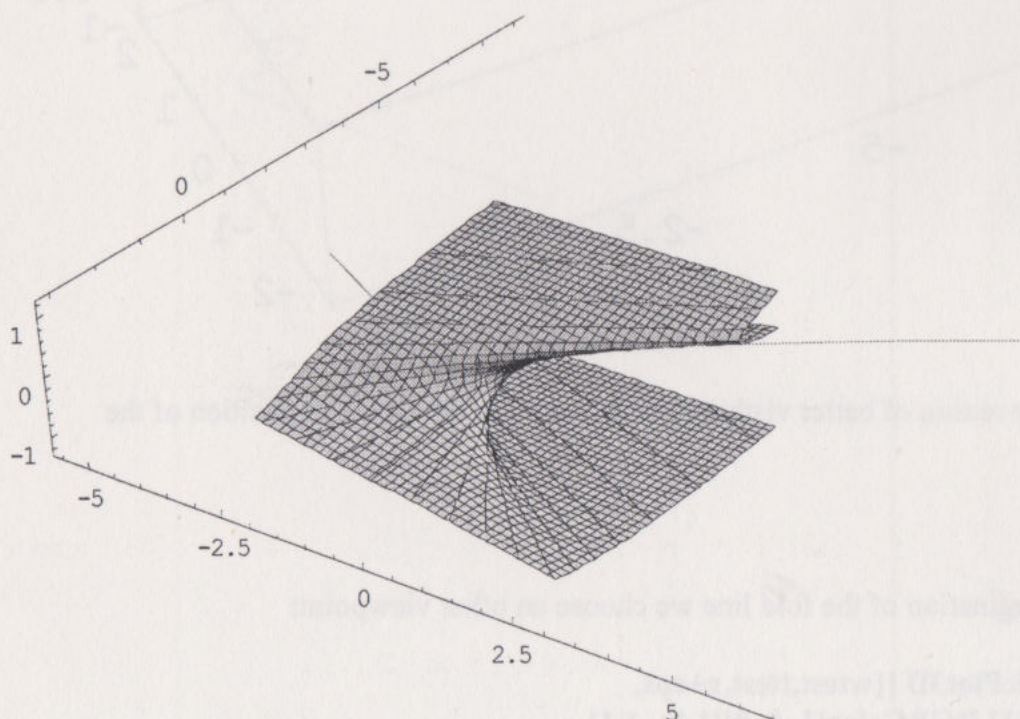
To get a better imagination of the fold line we choose an other viewpoint:

```
FLov= ParametricPlot3D [{wtest,ttest,x+eps,  
  {Thickness[0.001],RGBColor[1, 0, 0]}},{x,-1,1},  
  ViewPoint->{2.5, 1.5, 1.5}]
```



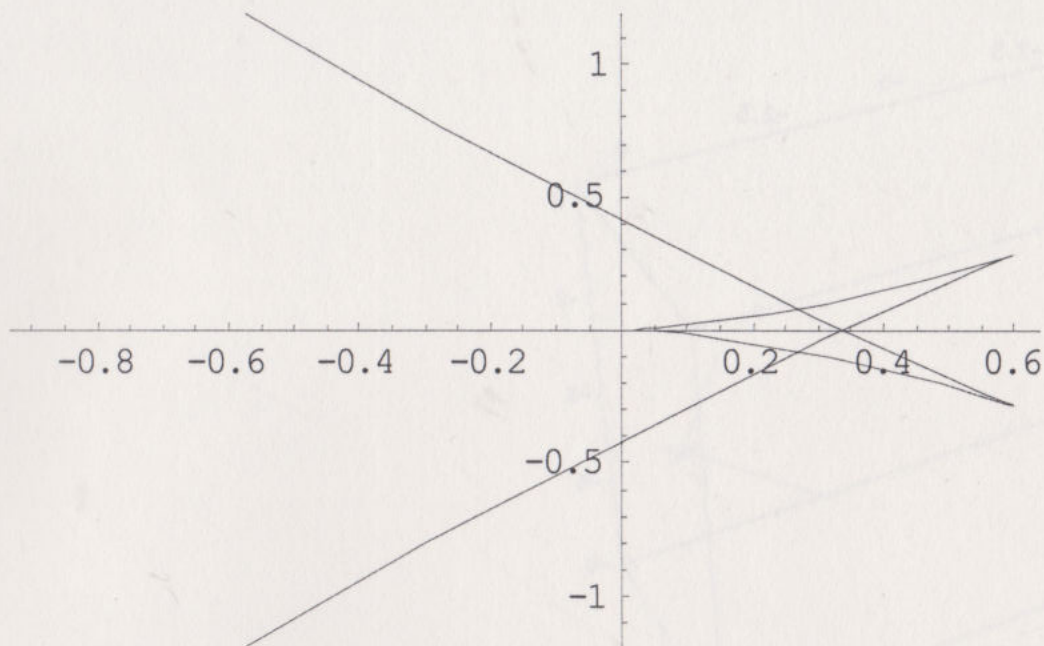
Now the composition::

Bildcompose1=Show[CriticalPoints,FLov]



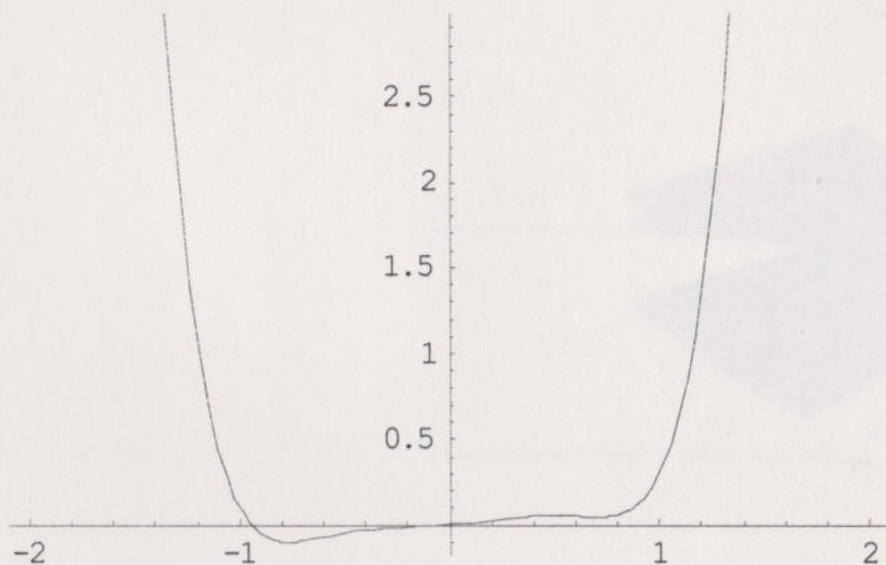
The following projection of the fold line onto w-t plane of the parameter space (u,v are constant) makes clear the notion of a butterfly:

InParameterspace=ParametricPlot[{wtest,ttest},{x,-1,1}]



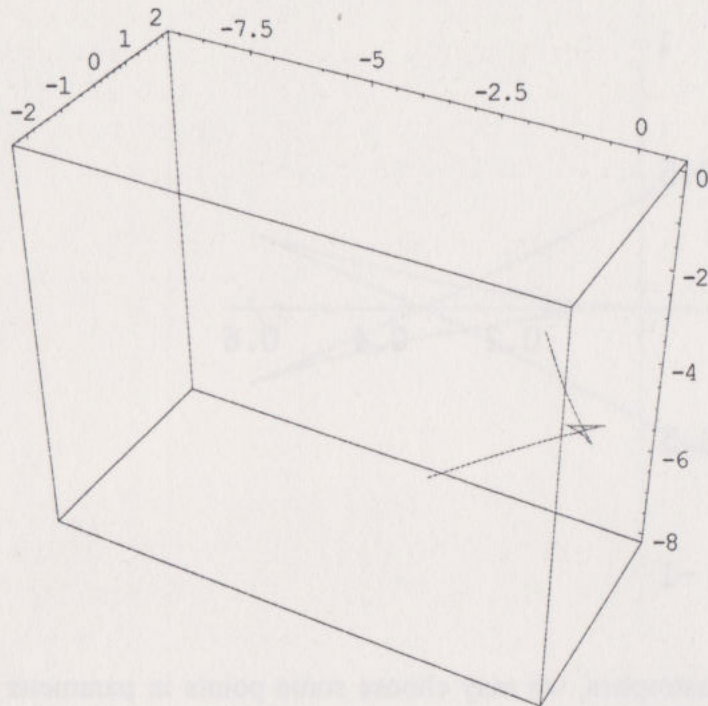
Here too, as with the two preceding catastrophes, we may choose some points in parameter space, to show the form of F laying "above" them:

Plot[F[x,-1,0,0.2,0.1],{x,-2,2}]

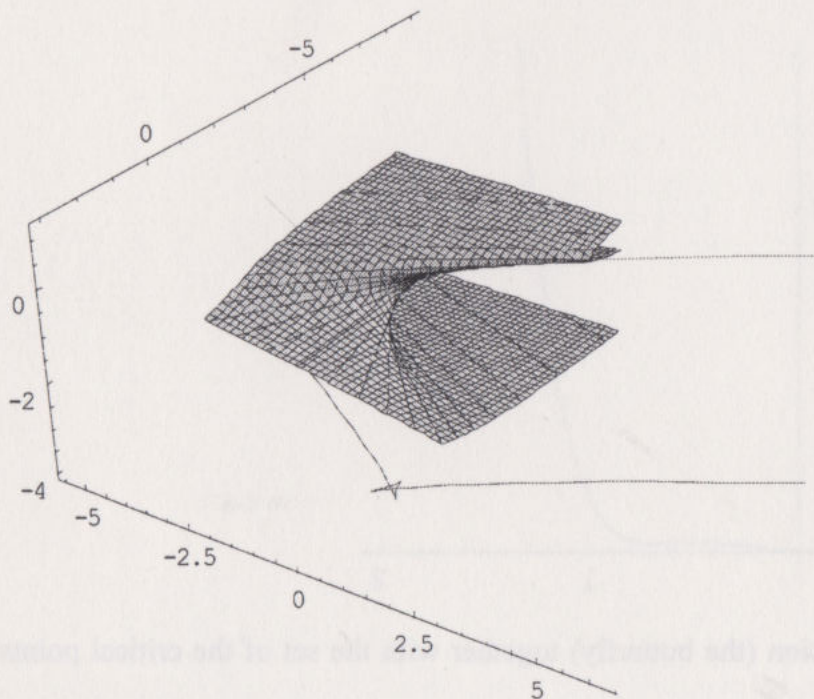


We want to visualize the projection (the butterfly) together with the set of the critical points (for u and v constant) in space:

**Projection3D=ParametricPlot3D [{wtest,ttest,-4,{Thickness[0.0005],
RGBColor[0, 0, 1]}},{x,-1,1}]**



Piccomp2=Show[CriticalPoints,fold,Projection3D]



At this place I would like to mention tree possibilities to spin around such a graphic (except with the “Viewpoint” statement). The first one is relatively slowly and can be realized with the standard version of Mathematica:

SpinShow[%]

The generated sequence of pictures can be moved by the animation possibilities of Mathematica. To do this you must mark the parentheses which contain all graphics and press “Strg” and “y” together.

The two following methods are essentially quicker and permit the rotation in all directions and other visual effect. The first program is called "MathLive", the second is a follower program called "Dynamic Visualizer". I have written the following statements in commentary symbols (* *) . You should execute these lines only if you have got these extra programs.

If you use "MathLive", the following two lines are needed without "(*" and "*)". The further handling can be found in its handbook. Only in short: The upper picture named Piccomp2 is saved here in a file named "Piccomp2.ts" and can be called from MathLive as 3-script-file and then be turned around.

```
(* Needs["Graphics`ThreeScript`"] *)
(* ThreeScript["Piccomp2.ts",Piccomp2] *)
```

For the "Dynamic-Visualizer" you use the following statements, if the program was installed in "c:\Dynvis". The picture "Piccomp2" is getting loaded and you can change it (turn it with pressed mouse key, magnify it with Strg and left mouse button, move with Shift- and left mouse key etc., see help button in this program).

```
SetDirectory["c:\Dynvis"]
Needs["DynamicVisualizer`"]
Visualize[Piccomp2]
```



7.5 The hyperbolic umbilic

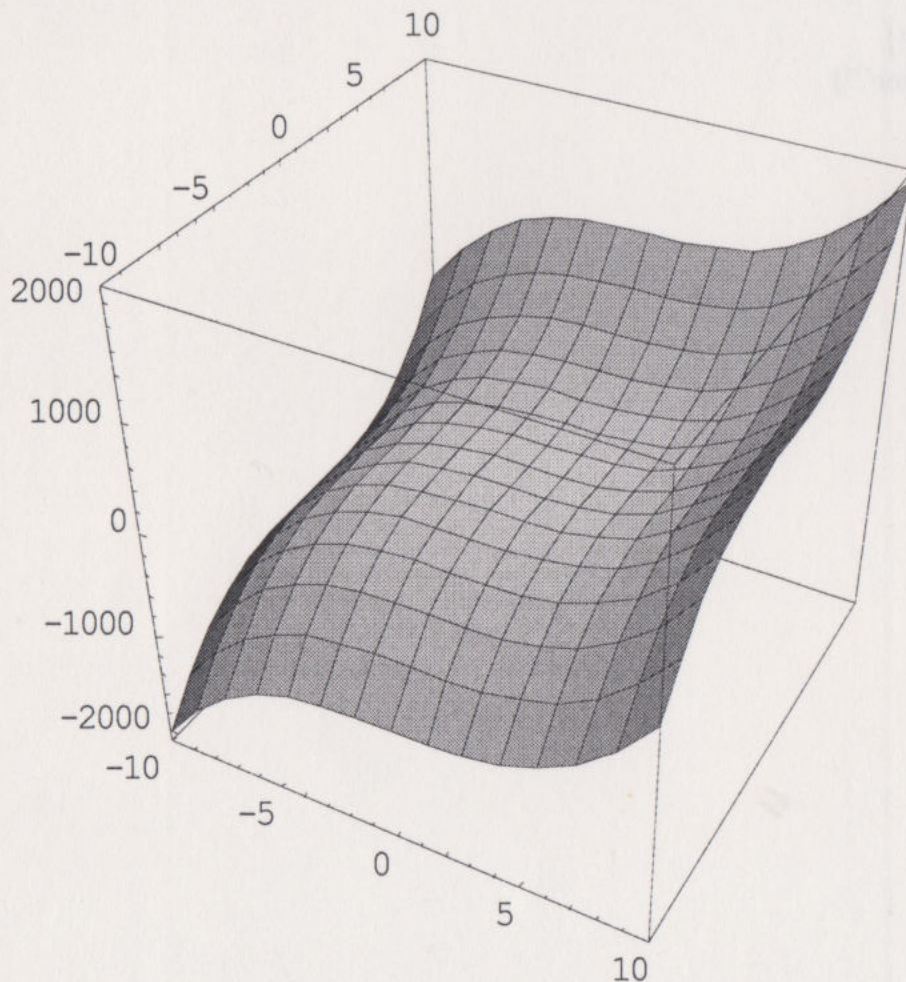
This elementary catastrophe is the universal unfolding of the function germ $f(x,y)=x^3+y^3$. The codimension of f is 3, and therefore the universal unfolding F of f has three unfolding parameters, which we call u , v and w .

```
Remove["Global`*"]
```

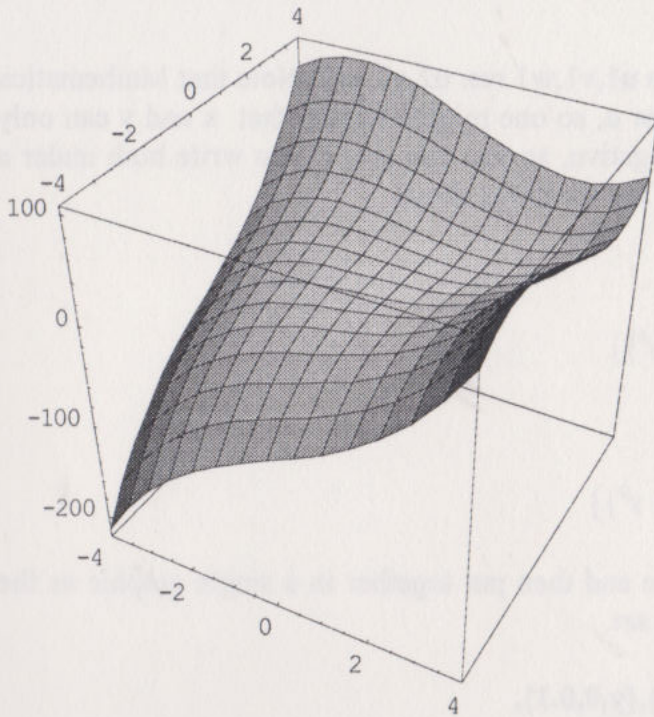
```
F[x_,y_,u_,v_,w_]:=x^3+y^3+u*x*y+v*x+w*y
```

We generate two graphics for F with fixed values form u,v and w , which we have chosen arbitrarily.

```
Plot3D[F[x,y,u,v,w]/.{u->0,v->1,w->1},{x,-10,10},{y,-10,10},BoxRatios->{1,1,1}];
```



```
Plot3D[F[x,y,u,v,w]/.{u->-5,v->1,w->1},{x,-4,4},{y,-4,4},BoxRatios->{1,1,1}];
```

Next we compute the gradient and the Hessian matrix of F , as well as its determinant:

```
Jacobi[functions_List, variables_List] := Outer[D, functions, variables]
```

```
{gradf}=Jacobi[{F[x,y,u,v,w]},{x,y}];
```

```
gradf
```

```
{v + 3 x^2 + u y, w + u x + 3 y^2}
```

```
H[s_,var_List]:=Map[D[Map[D[s,#]&,var],#]&,var]
```

```
Hesse=H[F[x,y,u,v,w]},{x,y}];
```

```
Hesse//MatrixForm
```

$$\begin{pmatrix} 6x & u \\ u & 6y \end{pmatrix}$$

```
dethesse=Det[Hesse]
```

```
-u^2 + 36 x y
```

The set of degenerated critical points is given by the solution of the equations $\text{grad } f=0$ and $\text{Det}[\text{Hesse}]=0$.

```
ausgCP=Solve[{gradf=={0,0},dethesse==0},{u,v,w}]
```

```
{ {v -> -3 (x^2 - 2 sqrt(x) y^(3/2)), w -> 3 (2 x^(3/2) sqrt(y) - y^2), u -> -6 sqrt(x) sqrt(y)},
  {v -> -3 (x^2 + 2 sqrt(x) y^(3/2)), w -> -3 (2 x^(3/2) sqrt(y) + y^2), u -> 6 sqrt(x) sqrt(y)} }
```

We store the two solutions in the variables $u1, v1, w1$ res. $u2, v2, w2$. Note that Mathematica has separated the two roots in the solution for u , so one might assume, that x and y can only be positive. But they both can as well be negative, as you can see, if you write both under a single root symbol in the third component of the solutions above.

```
{u1,v1,w1}={u,v,w}/.ausgCP[[1]]
```

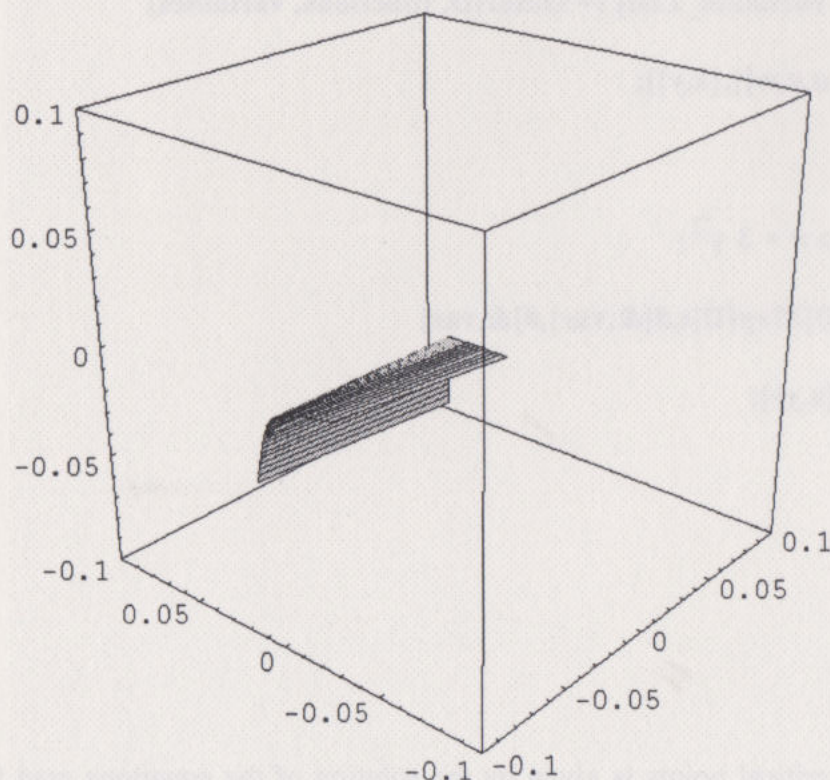
$$\{-6\sqrt{x}\sqrt{y}, -3(x^2 - 2\sqrt{x}y^{3/2}), 3(2x^{3/2}\sqrt{y} - y^2)\}$$

```
{u2,v2,w2}={u,v,w}/.ausgCP[[2]]
```

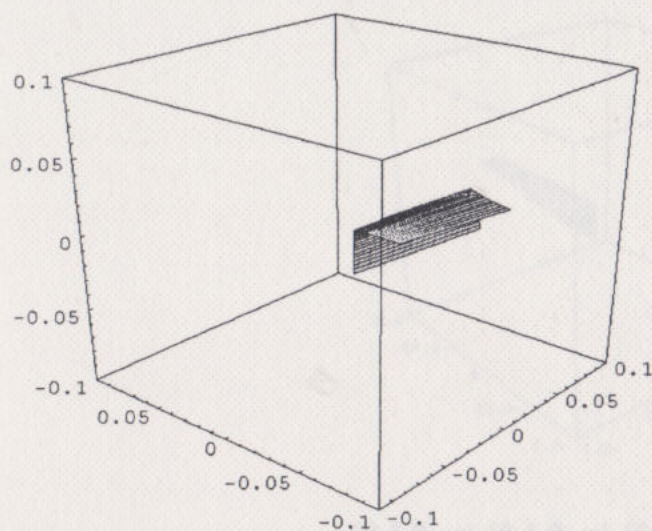
$$\{6\sqrt{x}\sqrt{y}, -3(x^2 + 2\sqrt{x}y^{3/2}), -3(2x^{3/2}\sqrt{y} + y^2)\}$$

Both solutions are shown in parameter space and then put together in a single graphic as the first part (x and y positive) of the bifurcation set.

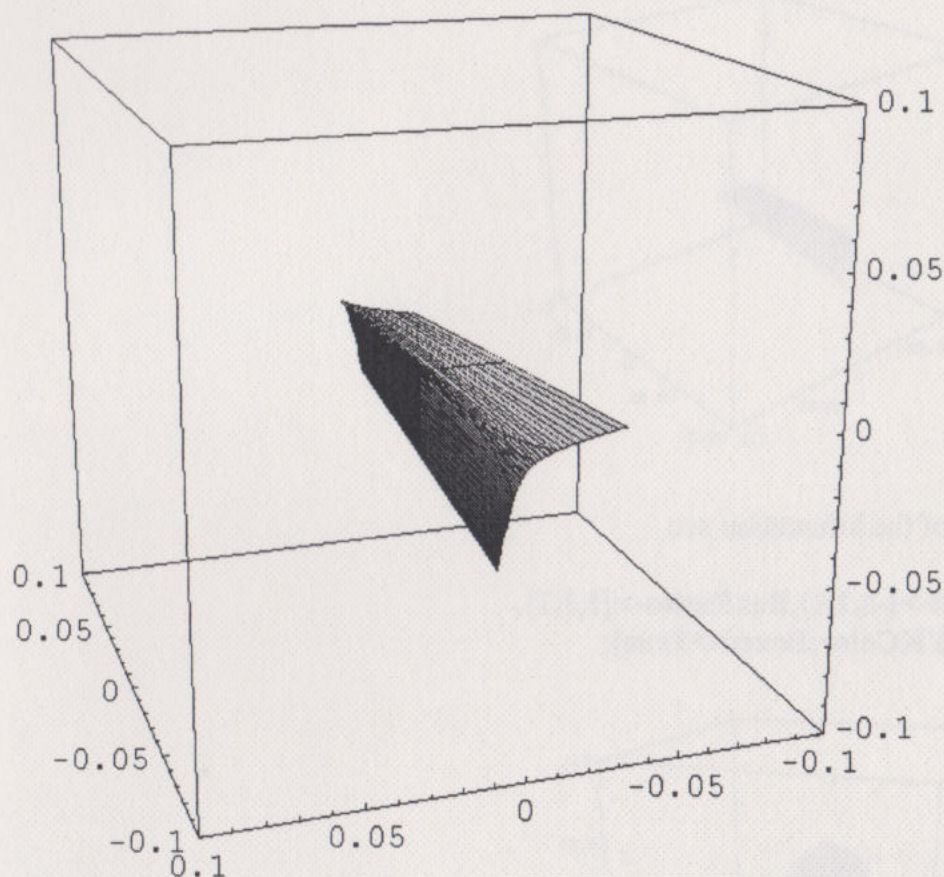
```
B1=ParametricPlot3D[{u1,v1,w1},{x,0,0.1},{y,0,0.1},  
ViewPoint->{-2.399, -2.053, 1.434},BoxRatios->{1,1,1},  
PlotRange->{{-0.1,0.1},{-0.1,0.1},{-0.1,0.1}}];
```



```
B2=ParametricPlot3D[{u2,v2,w2},{x,0,0.1},{y,0,0.1},  
ViewPoint->{-2.399, -2.053, 1.434},BoxRatios->{1,1,1},  
PlotRange->{{-0.1,0.1},{-0.1,0.1},{-0.1,0.1}}];
```

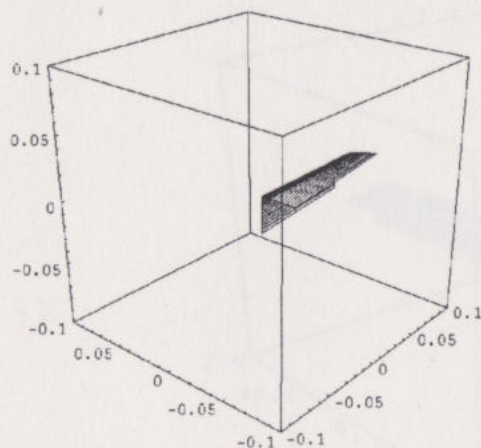



**Show[B1,B2,ViewPoint->{-3,1,1},BoxRatios->{1,1,1},
ColorOutput->CMYKColor;Boxed->True];**

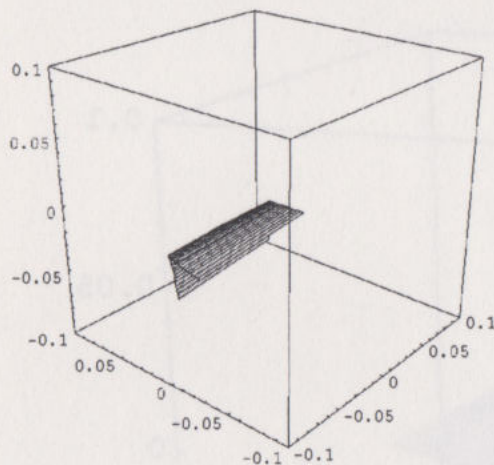


The same way we generate two graphics with x and y both negative

**B3=ParametricPlot3D[{u1,v1,w1},{x,-0.1,0},{y,-0.1,0},
ViewPoint->{-2.399, -2.053, 1.434},BoxRatios->{1,1,1},
PlotRange->{{-0.1,0.1},{-0.1,0.1},{-0.1,0.1}}];**

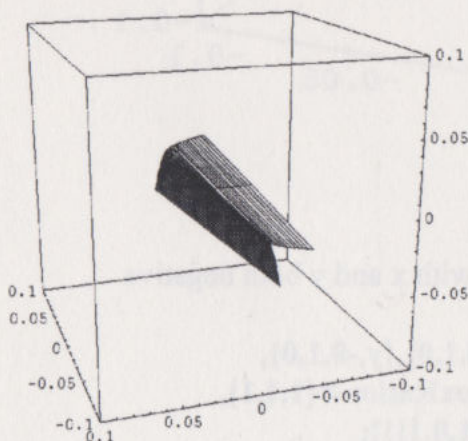


```
B4=ParametricPlot3D[{u2,v2,w2},{x,-0.1,0},{y,-0.1,0},
ViewPoint->{-2.399,-2.053,1.434},BoxRatios->{1,1,1},
PlotRange->{{-0.1,0.1},{-0.1,0.1},{-0.1,0.1}}];
```



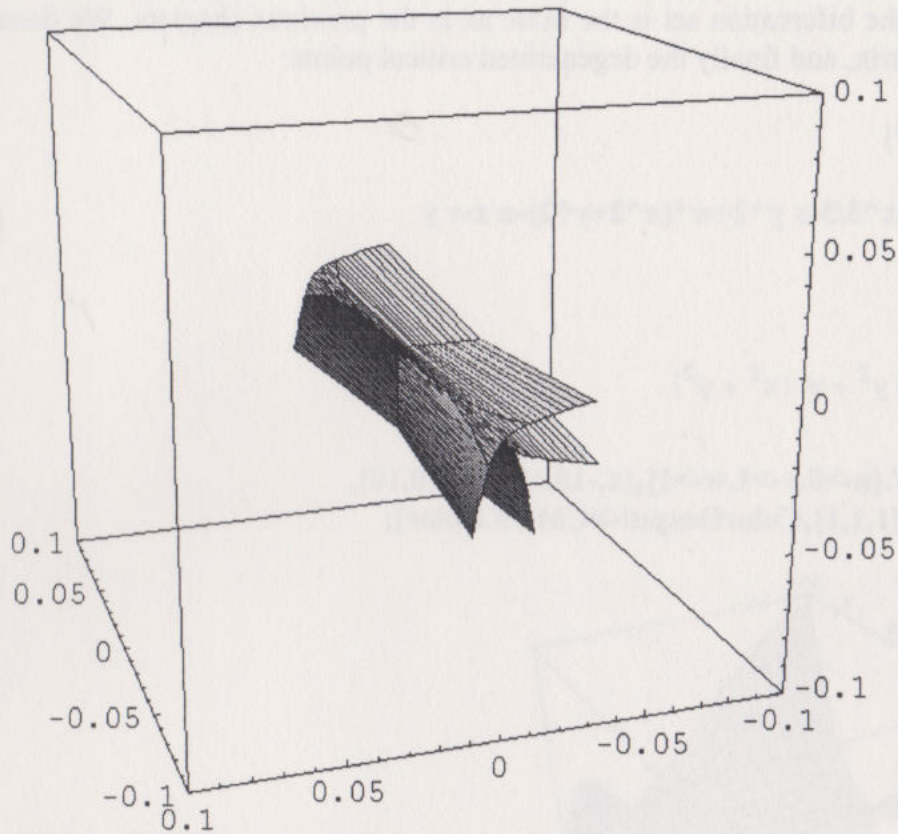
We get the second part of the bifurcation set:

```
Show[B3,B4,ViewPoint->{-3,1,1},BoxRatios->{1,1,1},
ColorOutput->CMYKColor;Boxed->True];
```



Putting all pictures together we receive the complete bifurcation set.

```
B=Show[B1,B2,B3,B4,ViewPoint->{-3,1,1},  
ColorOutput->CMYKColor,  
BoxRatios->{1,1,1},Boxed->True]
```



As an exercise, you may choose single points in different regions of the parameter space, which are separated by this set, and draw the corresponding sections of F .

7.6 The elliptical umbilic

Again we have a catastrophe of codimension 3. It has the equation:

$$F(x,y,u,v,w)=x^3/3-x y^2+w \cdot (x^2+y^2)-u x-v y.$$

Here I have chosen the names of the parameters in such a way, that it agrees with literature. The construction of the bifurcation set is the same as in the previous chapters. We determine gradient, Hessian matrix, and finally the degenerated critical points:

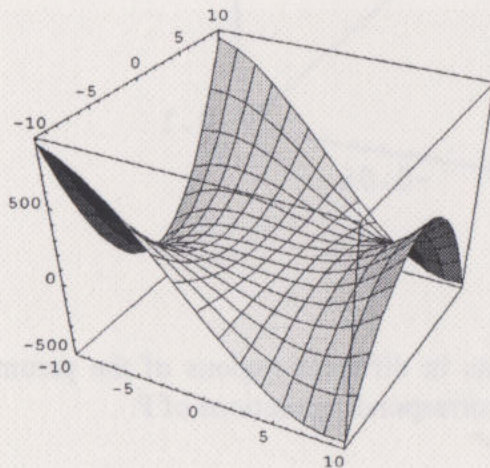
```
Remove["Global`*"]
```

```
F[x_,y_,u_,v_,w_]:=x^3/3-x y^2+w*(x^2+y^2)-u x-v y
```

```
F[x,y,u,v,w]
```

$$-u x + \frac{x^3}{3} - v y - x y^2 + w (x^2 + y^2)$$

```
Plot3D[F[x,y,u,v,w]/.{u->0,v->1,w->1},{x,-10,10},{y,-10,10},
BoxRatios->{1,1,1},ColorOutput->CMYKColor];
```



Gradient and Hessian matrix:

```
Jacobi[functions_List, variables_List] := Outer[D, functions, variables]
```

```
{gradf}=Jacobi[{F[x,y,u,v,w]},{x,y}];
```

```
gradf
```

$$\{-u + 2 w x + x^2 - y^2, -v + 2 w y - 2 x y\}$$

```
H[s_,var_List]:=Map[D[Map[D[s,#]&,var],#]&,var]
```

```
Hessian=H[F[x,y,u,v,w]},{x,y}];
```

```
Hessian//MatrixForm
```


$$\begin{pmatrix} 2w + 2x & -2y \\ -2y & 2w - 2x \end{pmatrix}$$

dethessian=Det[Hessian]

$$4w^2 - 4x^2 - 4y^2$$

We solve the system of equations for the degenerated critical points:

degCP=Solve[{gradf=={0,0},dethessian==0},{u,v,w}]

$$\left\{ \begin{aligned} & \{u \rightarrow x^2 - y^2 - 2x\sqrt{x^2 + y^2}, v \rightarrow -2(xy + y\sqrt{x^2 + y^2}), w \rightarrow -\sqrt{x^2 + y^2}\}, \\ & \{u \rightarrow x^2 - y^2 + 2x\sqrt{x^2 + y^2}, v \rightarrow 2(-xy + y\sqrt{x^2 + y^2}), w \rightarrow \sqrt{x^2 + y^2}\} \end{aligned} \right.$$

Then we give the values from the rule lists into variables:

{u1,v1,w1}={u,v,w}/.ausgCP[[1]]

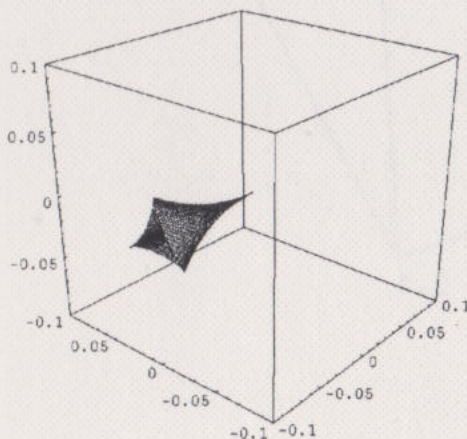
$$\{x^2 - y^2 - 2x\sqrt{x^2 + y^2}, -2(xy + y\sqrt{x^2 + y^2}), -\sqrt{x^2 + y^2}\}$$

{u2,v2,w2}={u,v,w}/.ausgCP[[2]]

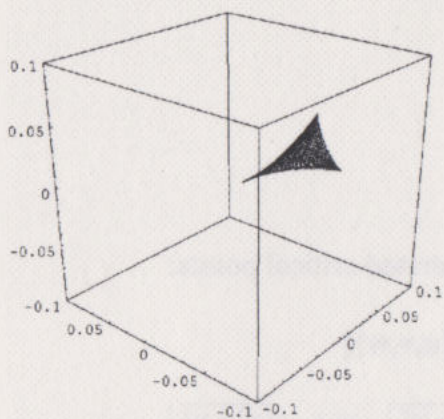
$$\{x^2 - y^2 + 2x\sqrt{x^2 + y^2}, 2(-xy + y\sqrt{x^2 + y^2}), \sqrt{x^2 + y^2}\}$$

Again we draw both parts separately and then in a single graphic the complete bifurcation set.

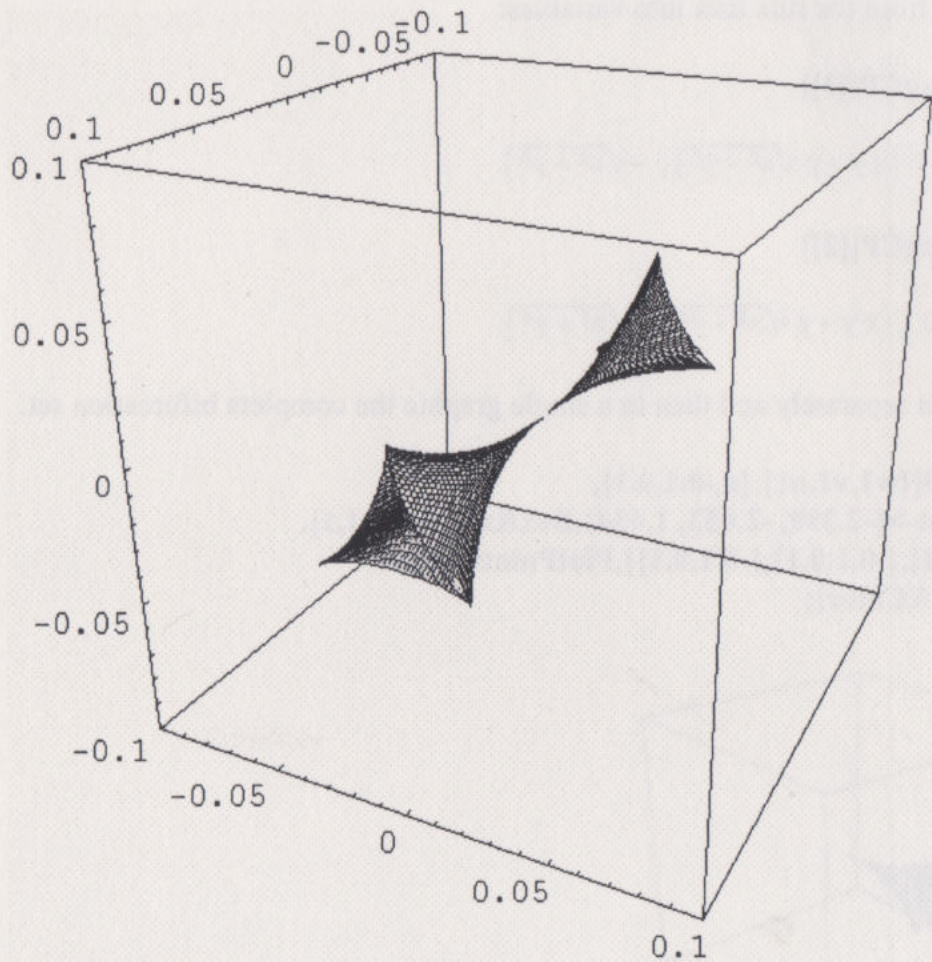
**Pic1=ParametricPlot3D[{w1,v1,u1},{x,-0.1,0.1},
{y,-0.1,0.1}, ViewPoint->{-2.399, -2.053, 1.434},BoxRatios->{1,1,1},
PlotRange->{{-0.1,0.1},{-0.1,0.1},{-0.1,0.1}},PlotPoints->50,
ColorOutput->CMYKColor];**



**Pic2=ParametricPlot3D[{w2,v2,u2},{x,-0.1,0.1},
{y,-0.1,0.1}, ViewPoint->{-2.399, -2.053, 1.434},BoxRatios->{1,1,1},
PlotRange->{{-0.1,0.1},{-0.1,0.1},{-0.1,0.1}},PlotPoints->50,
ColorOutput->CMYKColor];**



```
Show[Pic1,Pic2,ViewPoint->{2, 1, 1}];
```



7.7 The parabolic umbilic

It is given by the unfolding $F(x,y,w,t,u,v)=y^4+x^2\cdot y+u\cdot x^2+v\cdot y^2+w\cdot x+t\cdot y$ of $f(x,y)=y^4+x^2\cdot y$. Its codimension is 4, and we must be content to draw sections of the bifurcation set, where one of the parameters is fixed. The way to do this is quite analogous to the previous chapters. As an example, we generate one of the sections. A more detailed investigation of the parabolic umbilic is found in the book of Majthay (s. references).

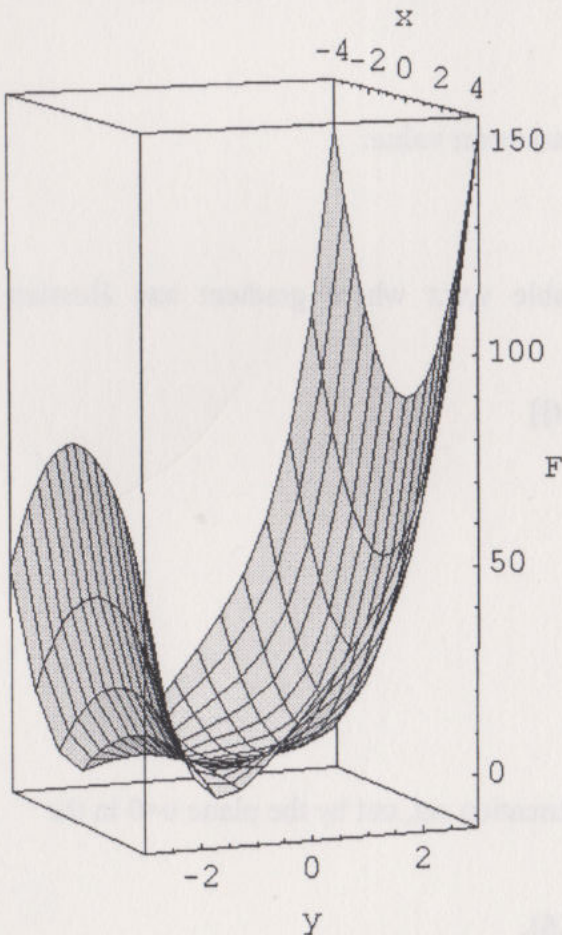
We remove all variables and functions and then define F:

```
Remove["Global`*"]
```

```
F[x_,y_,u_,v_,w_,t_]:=y^4+x^2*y+u*x^2+v*y^2+w*x+t*y
```

First we draw F itself with all four parameters having fixed values (see next statement). You may do some experiments with that values. If all 4 parameters are zero, then we just have the organization center of the parabolic umbilic. We choose as an example for the parameters u,w and t the value 1, for v the value 0:

```
Plot3D[F[x,y,u,v,w,t]/.{u->1,v->0,w->1,t->1},{x,-4,4},{y,-3,3},
  BoxRatios->{1,1,2}, ViewPoint->{12,-5,2.},Mesh->True,
  AxesLabel->{"x","y","F"},
  ColorOutput->CMYKColor];
```



For a graphic of the sections of the bifurcation set with the planes corresponding to an constant parameter, we need some preparations. First we calculate the gradient of F (by x and y):

Jacobi[functions_List, variables_List] := Outer[D, functions, variables]

{gradf}=Jacobi[{F[x,y,w,t,u,v]},{x,y}];

gradf

$\{w + 2 u x + 2 x y, t + x^2 + 2 v y + 4 y^3\}$

We now calculate the Hessian matrix and its determinant:

H[s_,var_List]:=Map[D[Map[D[s,#]&,var],#]&,var]

Hessian=H[F[x,y,w,t,u,v]},{x,y}];

Hessian//MatrixForm

$$\begin{pmatrix} 2u+2y & 2x \\ 2x & 2v+12y^2 \end{pmatrix}$$

dethessian=Det[Hessian]

$4 u v - 4 x^2 + 4 v y + 24 u y^2 + 24 y^3$

For the first section we choose for example $u=0$ as parameter value:

u=0;

We solve the system of equations for the variable v,w,t where gradient and Hessian determinant vanish:

degCP=Solve[{gradf=={0,0},dethessian==0},{v,w,t}]

$\left\{ \left\{ t \rightarrow -3x^2 + 8y^3, w \rightarrow -2xy, v \rightarrow -\frac{-x^2 + 6y^3}{y} \right\} \right\}$

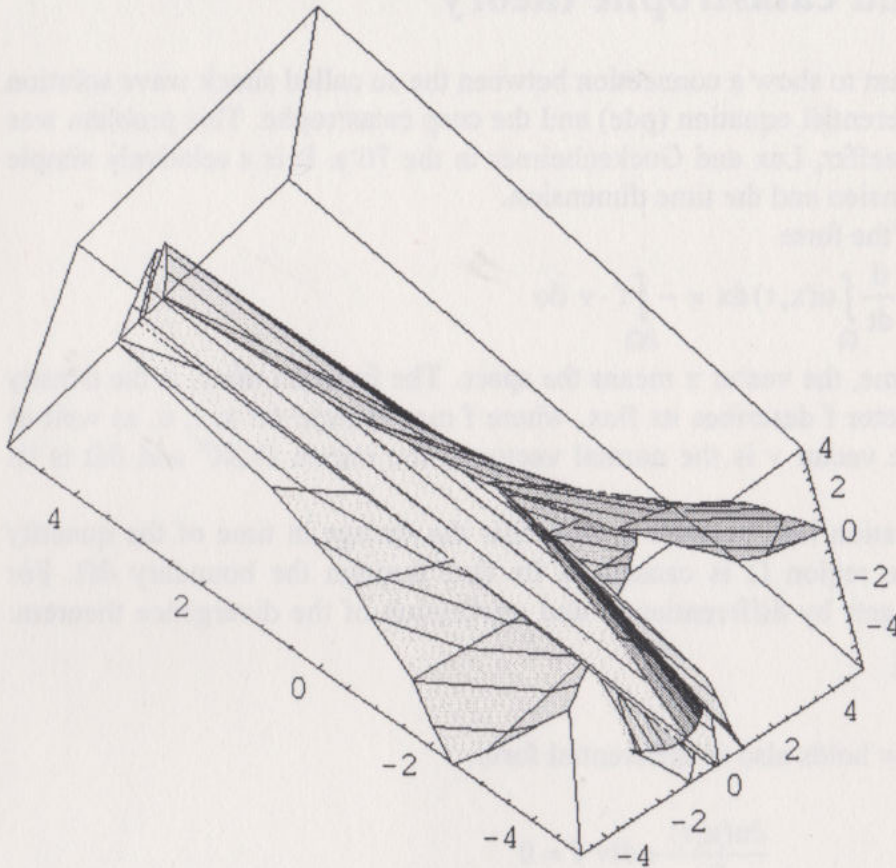
{v1,w1,t1}={v,w,t}/.degCP[[1]]//Simplify

$\left\{ -\frac{-x^2 + 6y^3}{y}, -2xy, -3x^2 + 8y^3 \right\}$

We get a parameterized (by x and y) graph of the bifurcation set, cut by the plane $u=0$ in the v,w,t-space

Picu0=ParametricPlot3D[{v1,w1,t1},{x,-5,5},{y,-5,5},


```
ViewPoint->{1., 1, -2}, BoxRatios->{2,1,1},
PlotRange->{{-5,5},{-5,5},{-5,5}},PlotPoints->50,
ColorOutput->CMYKColor];
```



If you have got the program "Dynamic Visualizer" you may give in the following three input lines. This enables you to turn around the picture as you want. Otherwise please do not give in these lines.

```
SetDirectory["c:\Dynvis"]
```

```
"c:\Dynvis"
```

```
Needs["DynamicVisualizer`"]
```

```
Visualize[Picu0];
```

At the end of this chapter, please do some examples with other values of the parameters.

8 Shock waves and catastrophe theory

As an application, we want to show a connection between the so called shock wave solution of a quasilinear partial differential equation (pde) and the cusp catastrophe. This problem was treated by Golubitsky, Schaeffer, Lax and Guckenheimer in the 70's. It is a relatively simple example in one space dimension and the time dimension.

Look at physical laws of the form

$$\frac{d}{dt} \int_{\Omega} u(x, t) dx = - \int_{\partial\Omega} f \cdot \nu \, d\sigma$$

The variable t means the time, the vector x means the space. The function $u(x, t)$ is the density of a physical entity, the vector f describes its flux, where f may depend on x , t , u , as well as the partial derivatives. The vector ν is the normal vector to the region $\Omega \subset \mathbb{R}^n$ and $\partial\Omega$ is its boundary.

Such a law is a conservation law, because it says, that the change in time of the quantity ("mass") $\int u \, dx$ in a space region Ω is caused by its flux through the boundary $\partial\Omega$. For differentiable u and f one gets by differentiation and application of the divergence theorem:

$$\int_{\Omega} \left\{ \frac{\partial}{\partial t} u(x, t) + \operatorname{div} f \right\} dx = 0$$

Since Ω is arbitrary, this law holds also in differential form:

$$\frac{\partial u(x, t)}{\partial t} + \operatorname{div} f = 0$$

We use in our mathematical model the following simplifications: x be only one space dimension, and thus $\operatorname{div} f = \partial_x f(u(x, t))$. Such restrictions are not of physical interest, since shock phenomena are described by systems of partial differential equations, but are used here as simplification of our approach.

Thus we start with the partial differential equation of the form

$$\frac{\partial u(x, t)}{\partial t} + \frac{\partial f(u(x, t))}{\partial x} = 0$$

or equivalently, writing $\frac{\partial f}{\partial u} = f'(u) = a(u)$, with the differential equation

$$\frac{\partial u(x, t)}{\partial t} + a(u) \cdot \frac{\partial u(x, t)}{\partial x} = 0$$

The initial value problem is: Given $u_0(x) = u(x, 0)$, find $u(x, t)$ for all $t > 0$. Solutions of this pde may lead to discontinuities, even when we have continuous initial conditions. We call those differentiable curves in the x - t plane, along which discontinuities occur, shock curves or shock waves. From physical reasons, these have to fulfill additional conditions (jump condition, entropy condition), which we do not declare here. (For details see reference: Lax; Golubitsky and Schaeffer).

The occurrence of shock waves in front of an accelerated piston in a tube filled with gas is a

physical realization of such a model. The cusp catastrophe, whose connection to this shock wave phenomenon will be shown later, has the name “Riemann-Hugoniot catastrophe“, according to Thom.

Our pde is a quasilinear partial differential equation of first order (quasilinear means, that it is linear in the derivatives of highest order of the dependent variables). If there exists a solution $u(x,t)$ for the mentioned pde, then the vector field $(a(u), 1, 0)$ is tangent to the graph S of u , since the tangent space $T_p S$ to S at a point $p=(x,t,u(x,t))$ is spanned by the vectors $(1, 0, \frac{\partial u(x,t)}{\partial x})$ and $(0, 1, \frac{\partial u(x,t)}{\partial t})$:

$$T_p S = \left\{ \alpha \cdot (1, 0, \frac{\partial u(x,t)}{\partial x}) + \beta \cdot (0, 1, \frac{\partial u(x,t)}{\partial t}) \right\}; \alpha, \beta \in \mathbb{R}$$

Taking $\alpha=a(u)$ and $\beta=1$, one gets, since $\frac{\partial u(x,t)}{\partial t} + a(u) \cdot \frac{\partial u(x,t)}{\partial x} = 0$, the vector field $V=(a(u), 1, 0)$, being tangent to S . The integral curve $c(s)$, going through a fixed point $(x_0, t_0, u(x_0, t_0))$, is a line, since the third component of V vanishes and the others depend only on u . It can be written in the Form $c(s) = (x_0, t_0, u(x_0, t_0)) + s \cdot (a(u(x_0, t_0)), 1, 0)$. This can also be expressed in the following form: $c(s) = (x_0 + s \cdot a(u(x_0, t_0)), s + t_0, u(x_0, t_0))$. The last form shows us, how we can choose a parameterization Ψ of graph of $u(x,t)$:

$$\Psi(y,s) = (y + s \cdot a(u_0(y)), s, u_0(y)) , \text{ with } u(y,0) = u_0(y)$$

Let's visualize these things with Mathematica.

Remove["Global`*"]

You can give in the pde for example in the following form:

$$\text{pde} = D[u[x, t], t] + a[u] \cdot D[u[x, t], x] == 0$$

This is “InputForm“. With the menu sequence “Cell“, “ConvertTo“, and then “StandardForm” or “TraditionalForm“ you get the pde in one of the following forms, which you also can give in directly with help of the palettes:

$$\text{pde} = \partial_t u[x, t] + a[u] \partial_x u[x, t] == 0$$

or

$$\text{pde} = \frac{\partial u(x, t)}{\partial t} + a(u) \frac{\partial u(x, t)}{\partial x} == 0$$

In all cases you get (after pressing “Shift“ and “Return“) the answer:

$$u^{(0,1)}[x, t] + a[u] u^{(1,0)}[x, t] == 0$$

We solve the pde:

L=DSolve[pde,u[x,t],{x,t}]/FullSimplify

$$\left\{ \left\{ u[x, t] \rightarrow C[1] \left[t - \frac{x}{a[u]} \right] \right\} \right\}$$

In the solution we have an arbitrary function in $(t-x/a)$, which is named $C[1]$ by Mathematica. For the initial value problem $u(x,0)=u_0(x)$ we change the notation:

L=L/.{C[1]->u0}

$$\left\{ \left\{ u[x, t] \rightarrow u_0 \left[t - \frac{x}{a[u]} \right] \right\} \right\}$$

Moreover its usual in literature to give the argument not in the form $t-x/a(u)$, used by Mathematica, but in the form $x-a(u) \cdot t$.

We show, that $u(x,t)=u_0(x-a(u) \cdot t)$ is ended a solution of the pde

u[x_,t_]:=u0[x-a[u]*t]

pde

True

Now we look at the above parameterization of the graph of u .

$\Psi[y, s] := \{y + s * a[u_0[y]], s, u_0[y]\}$

For a concrete initial value problem, we shall show, how the parameterization of the graph can be visualized in a graphic. We choose $f(u)=u^2/2$ and $u(x,0)=u_0(x)=e^{-x^2}$.

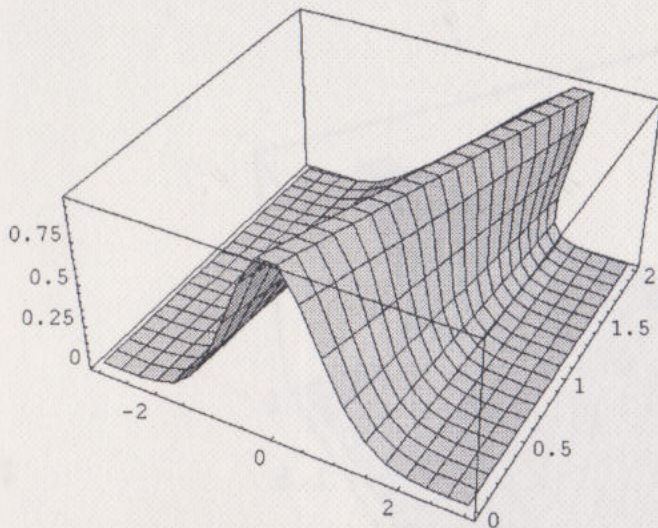
f[u_]:=u^2/2

a[u_]:=f'[u]

u0[x_]:=Exp[-x^2]

Needs["Graphics`Master`"]

**W = ParametricPlot3D[Evaluate[Ψ[y, s]], {y, -3, 3}, {s, 0, 2},
BoxRatios → {1, 1, 0.5}];**



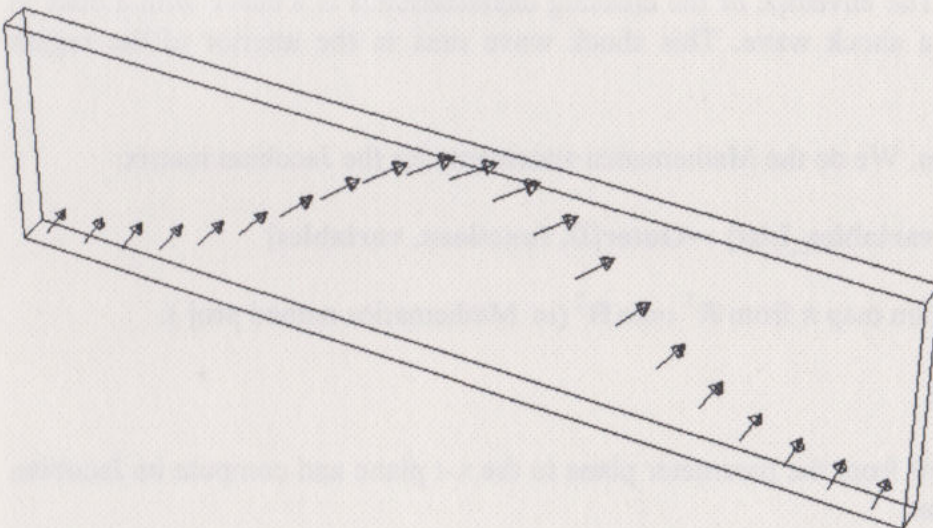
Let's draw the tangent vector field V (only the first row and adding $\text{eps}=0.3$ for the reason of visibility).

```
<<Graphics`PlotField3D`
```

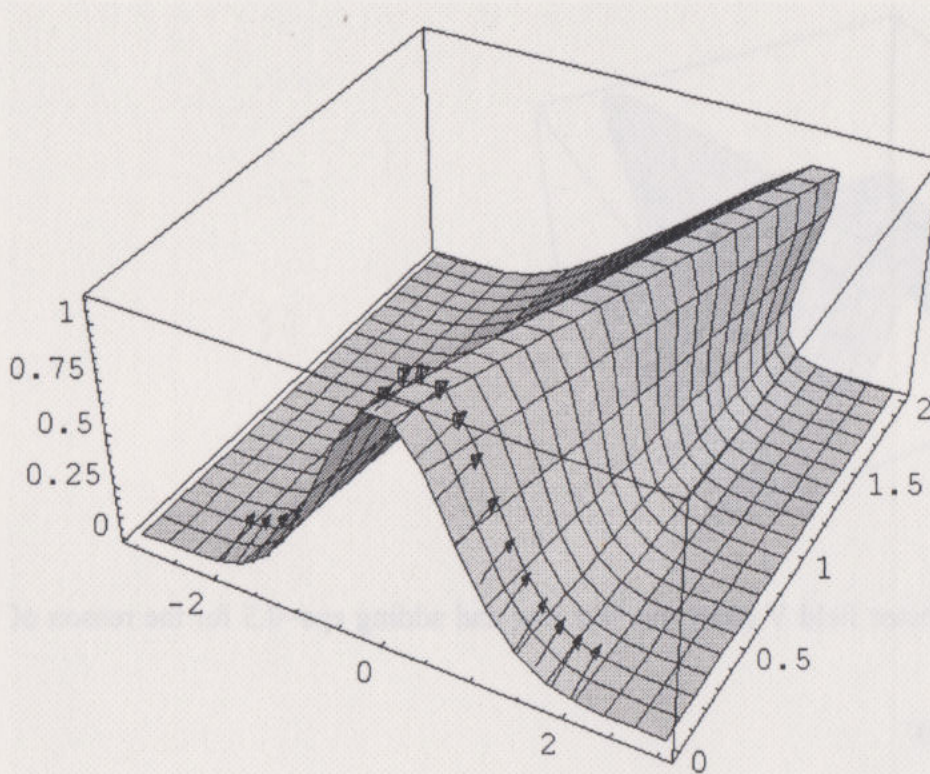
```
eps=0.3;
```

```
vlist = Table[{x,0,u0[x]}, {a[u0[x]],1,0+eps}},  
             {x, -2,2,0.2}];
```

```
vf=ListPlotVectorField3D[vlist,VectorHeads->True,  
                        ScaleFactor->0.3];
```



```
Show[W,vf];
```

Now we look at the map Ψ . Our interest is to determine the beginning of the folding, since at such points the solution isn't one-one, and here starts a new phenomenon:

If we take for example a long cylinder, in which an accelerated piston is moving, and if u is the density of the enclosed medium (gas, fluid), then we get after some time the phenomenon of the formation of a shock wave. The "world lines" (characteristics) of the particles in the x - t plane are straight lines, which have different inclinations against the x -axes under the influence of the piston. The envelope of the crossing characteristics is a curve with a cusp in the formation point of a shock wave. This shock wave runs in the interior of the region defined by the envelope.

Back to our computations. We do the Mathematica statements for the Jacobian matrix:

Jacobi[functions_List, variables_List] := Outer[D, functions, variables]

We now define a projection map π from $\mathbb{R}^3$ onto $\mathbb{R}^2$ (in Mathematica named **proj**):

proj[{l_, m_, n_}] := {l, m}

Then we form the map $\pi \circ \Psi$ from the parameter plane to the x - t plane and compute its Jacobian matrix and its determinant.

proj[Ψ[y, s]]

$\{e^{-y^2} s + y, s\}$

J=Jacobi[proj[Ψ[y,s]],{y,s}]

J//MatrixForm

$$\begin{pmatrix} 1 - 2 e^{-y^2} s y & e^{-y^2} \\ 0 & 1 \end{pmatrix}$$

Det[J]

$$1 - 2 e^{-y^2} s y$$

We compute the points in parameter space where the determinant vanishes:

kp=Solve[Det[J]==0,{s}]

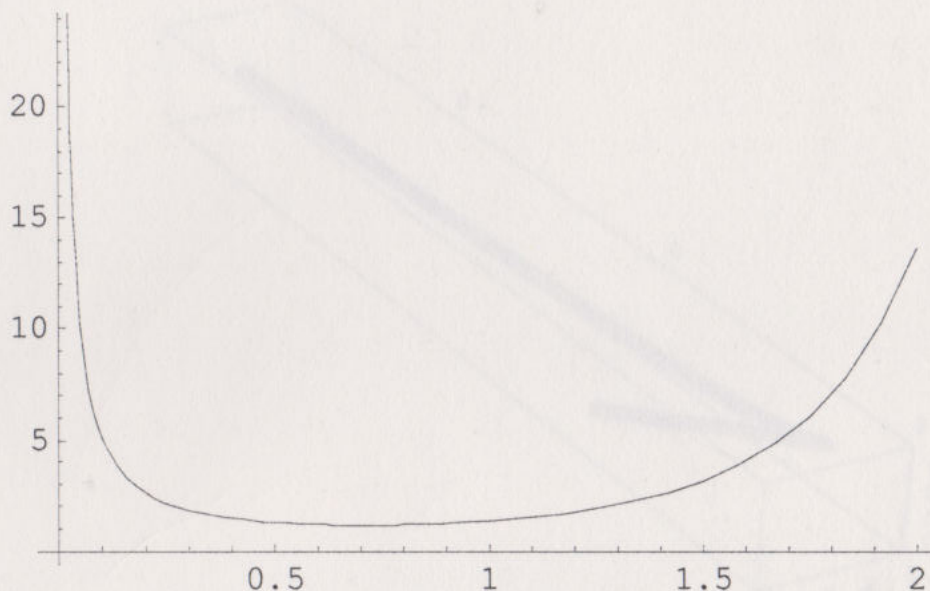
$$\left\{ \left\{ s \rightarrow \frac{e^{y^2}}{2 y} \right\} \right\}$$

s=s/.Flatten[kp]

$$\frac{e^{y^2}}{2 y}$$

We receive a curve in parameter space , which we now draw:

ParametricPlot[{y,s},{y,0.01,2}]

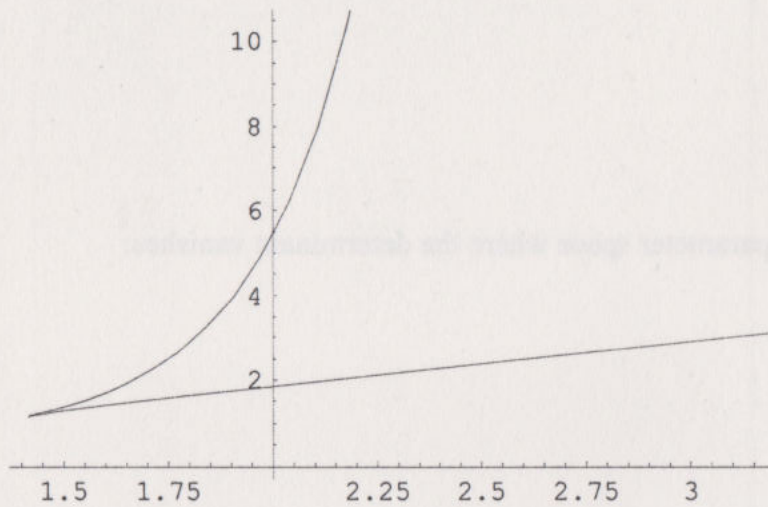


The image under Ψ of this curve in the x-t plane is now drawn too:

$\Psi[y,s]$

$$\left\{ \frac{1}{2 y} + y, \frac{e^{y^2}}{2 y}, e^{-y^2} \right\}$$

```
ParametricPlot[{Ψ[y,s][[1]],Ψ[y,s][[2]]} //Evaluate,{y,0.01,2}];
```

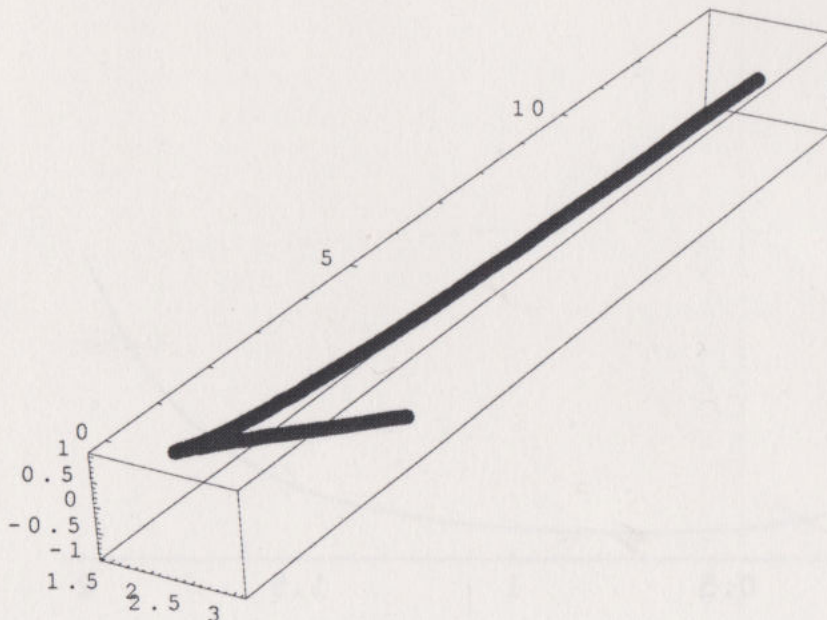


We want to show it in 3 space:

```
he=Evaluate[Flatten[{proj[Ψ[y,s]],0}]];
```

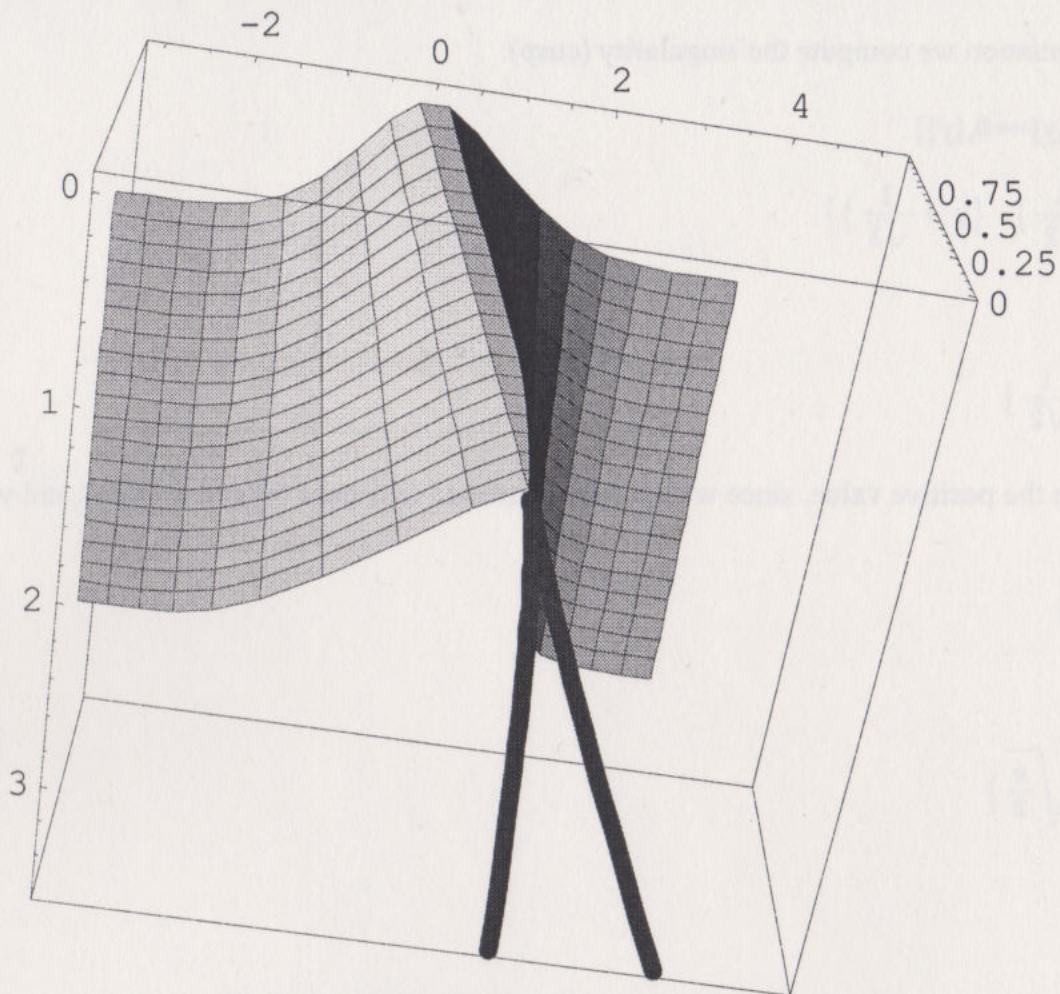
```
he2=Append[he,{Thickness[0.02],RGBColor[0,0,1]}];
```

```
spc=ParametricPlot3D[Evaluate[he2],{y,0.1,2}];
```

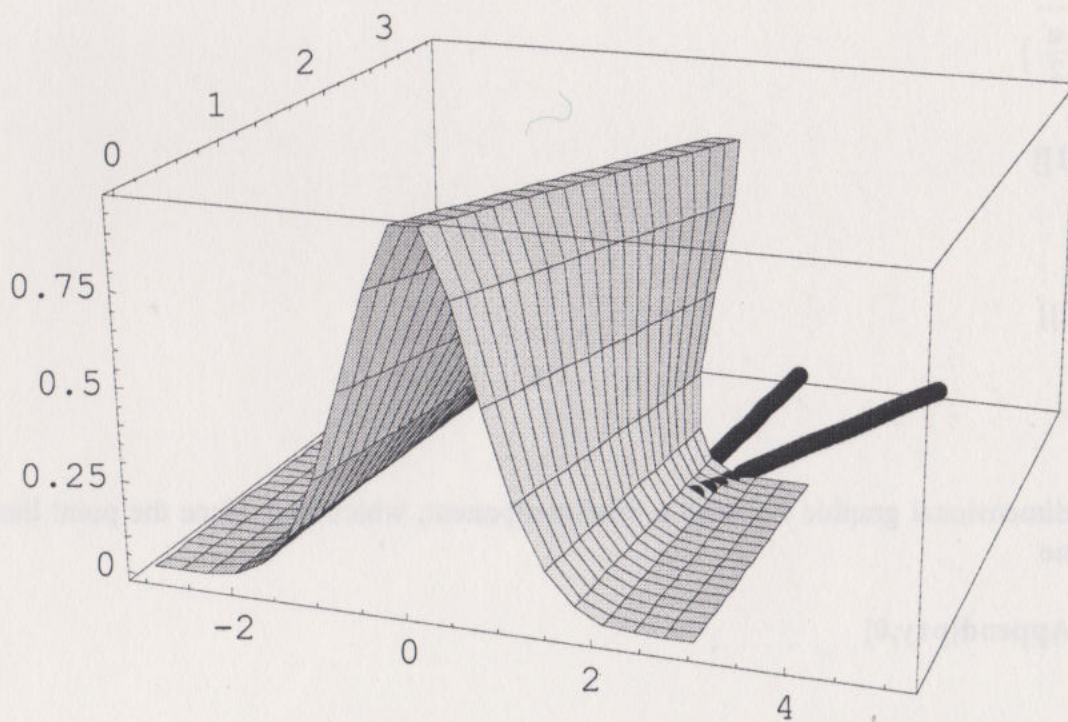


Now we show this cusp together with the graph of u , seen from a point „under“ the surface, so that both objects can be seen:

```
Show[W,spc,ViewPoint->{-0.231, -1.481, -2.872}]
```

`Show[W,spc,ViewPoint->{1,-3,1}];`



By differentiation we compute the singularity (cusp):

Solve[D[s,y]==0,{y}]

$$\left\{ \left\{ y \rightarrow -\frac{1}{\sqrt{2}} \right\}, \left\{ y \rightarrow \frac{1}{\sqrt{2}} \right\} \right\}$$

y=y/.%

$$\left\{ -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\}$$

We choose the positive value, since we are only interested in times $t > 0$ and for $t > 0$ s and y are positive.

yp=y[[2]];

s/.y->yp

$$\left\{ -\sqrt{\frac{e}{2}}, \sqrt{\frac{e}{2}} \right\}$$

sp=s[[2]];

We determine the coordinates of the image point and call these xm res. tm

pxy=proj[Ψ[yp,sp]]

$$\left\{ \sqrt{2}, \sqrt{\frac{e}{2}} \right\}$$

xm=pxy[[1]]

$$\sqrt{2}$$

tm=pxy[[2]]

$$\sqrt{\frac{e}{2}}$$

For the 3 dimensional graphic we need a third component, which is 0, since the point lies in the x-t plane

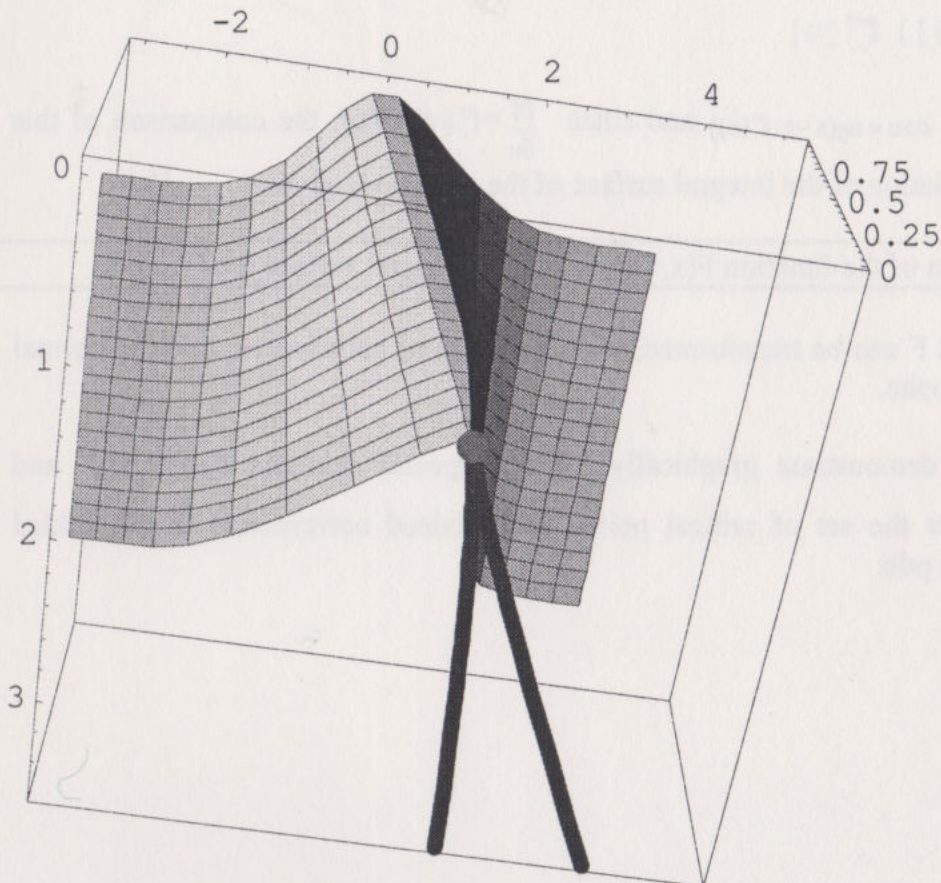
p3dproj=Append[pxy,0]

$$\{\sqrt{2}, \sqrt{\frac{e}{2}}, 0\}$$

```
cb=Graphics3D[{RGBColor[1,0,0],PointSize[0.04],
  Point[p3dproj]}}];
```

Now we draw all together

```
Show[W,spc,cb,ViewPoint->{-0.231, -1.481, -2.872}]
```



At physical interpretation of the problem, this cusp curve defines the region in x - t plane within which the shock wave runs. In Physics some other conditions for its characterization play a roll, e.g. the so called “jump condition” and the “entropy condition” in which we are not interested here (see reference). What we want to show is, that this cusp actually has a connection to Thom’s cusp catastrophe. We examine the following connection:

According to P. Lax (s. reference) to the solution $u(x,t)=u_0(x-a(u)\cdot t)$ of the pde the following function $F(x,t,u)$ is defined:

```
Remove["Global`*"]
```

```
F[x_,t_,u_]:=t*(u*f'[u]-f[u])+psi[x-f'[u]*t]
```

$\psi[y_]:=Integrate[u0[x],\{x,0,y\}]$

We compute the partial derivative of F by u:

$D[F[x,t,u],u]$

$t u f''[u] - t u0[x - t f'[u]] f''[u]$

Simplify[%]

$t (u - u0[x - t f'[u]]) f''[u]$

It follows $\frac{\partial F(x,t,u)}{\partial u} = 0 \Leftrightarrow u = u_0(x - t \cdot f'(u))$ and since $\frac{\partial f}{\partial u} = f'(u) = a(u)$, the comparison of this condition with the calculation of the integral surface of the pde above gives :

The set of singular points of the function $F(x,t,u)$ equals the integral surface S of the pde.

We intend to show, that F can be transformed, after a change of coordinates, into the normal form of the cusp catastrophe.

But first we want to demonstrate graphically for our special example $f(u) = 0.5 \cdot u^2$ and $u(x,0) = u_0(x) = e^{-x^2}$, that the set of critical points of F indeed corresponds to the folded solution surface S of the pde:

$f[u_]:=u^2/2$

$a[u_]:=f'[u]$

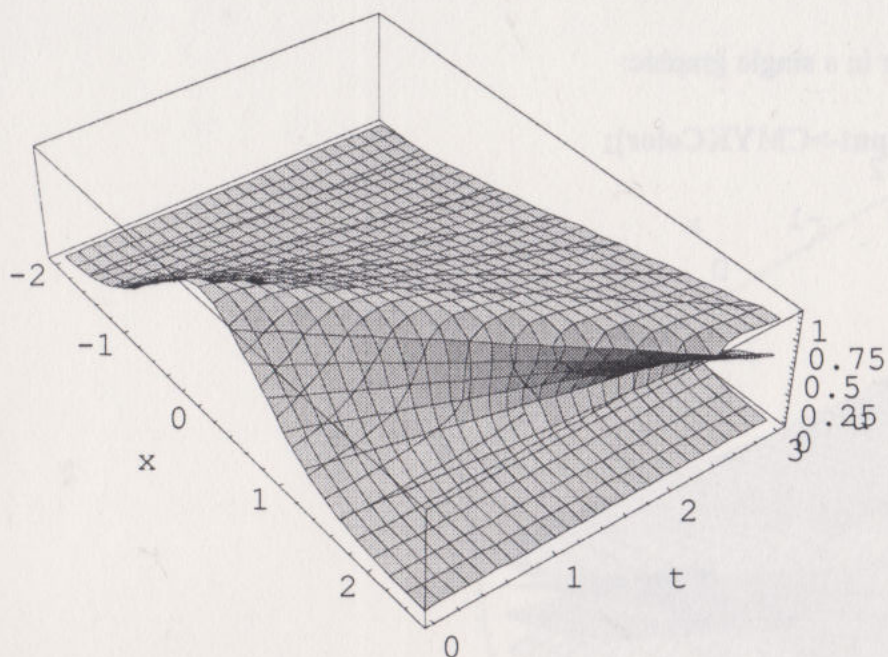
$u0[x_]:=Exp[-x^2]$

$u0[x-a[u]*t]$

<<Graphics`ContourPlot3D`

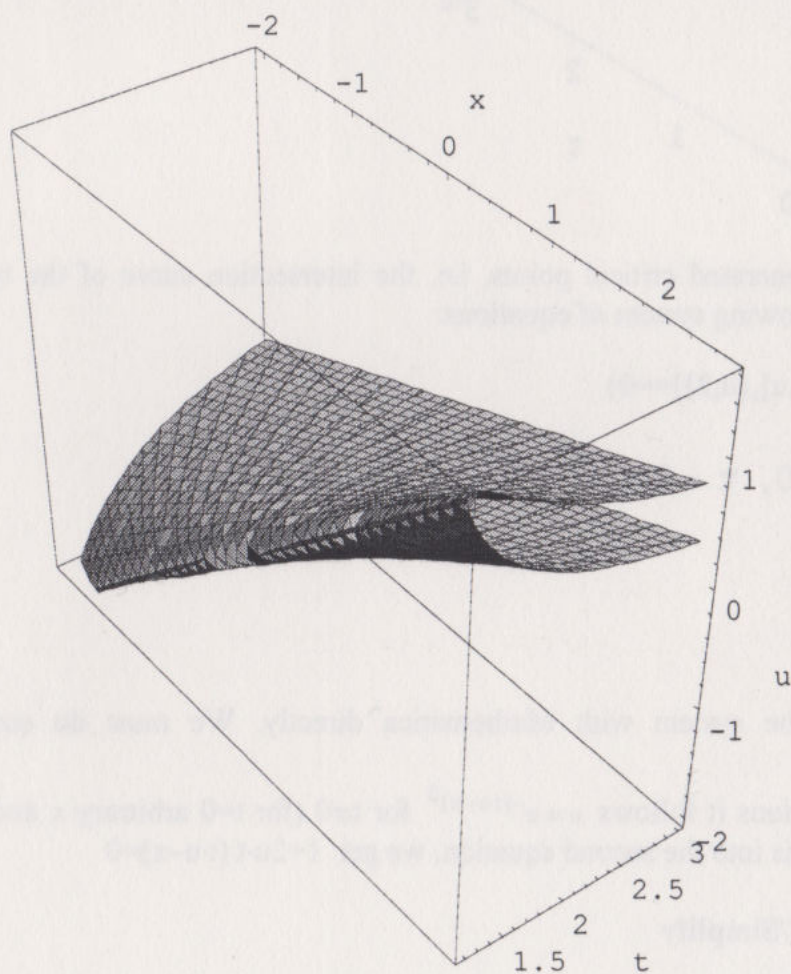
We draw the set of the critical point of F :

**CP1=ContourPlot3D[D[F[x,t,u],u], {x,-2,2.5},{t,0.01,3},{u,-2,2},
PlotPoints->6,ViewPoint->{2.0,-1.5,2},Axes->True,AxesLabel->{"x","t","u"}];**



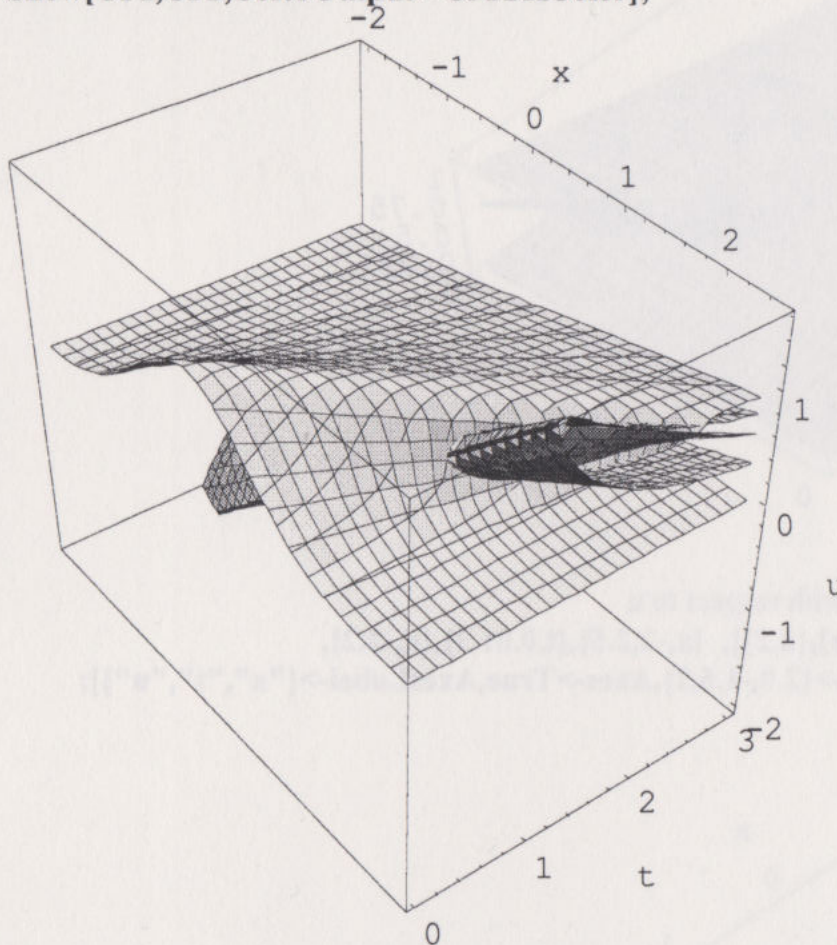
We draw the second derivative with respect to u

```
CP2=ContourPlot3D[D[F[x,t,u],{u,2}], {x,-2,2.5},{t,0.01,3},{u,-2,2},
  PlotPoints->7,ViewPoint->{2.0,-1.5,2},Axes->True,AxesLabel->{"x","t","u"}];
```



Now we draw it all together in a single graphic:

Show[CP1,CP2,ColorOutput->CMYKColor];



In order to determine the degenerated critical points, i.e. the intersection curve of the two surfaces, we must solve the following system of equations:

sys={D[F[x,t,u],u]==0,D[F[x,t,u],{u,2}]==0}

$$\left\{ -e^{-(tu-x)^2} t + tu == 0, t + 2e^{-(tu-x)^2} t^2 (tu-x) == 0 \right\}$$

We clear all values

Clear[u,x,t,u1,u2,x1,x2]

It is not possible to solve the system with Mathematica directly. We must do some transformations:

From the first of the two equations it follows $u = e^{-(tu-x)^2}$ for $t \neq 0$ (for $t=0$ arbitrary x and u solve the equation). Inserting this into the second equation, we get $1+2u \cdot t(tu-x)=0$

L=Solve[1+2u t (t u -x)==0,u]//Simplify

$$\left\{ \left\{ u \rightarrow \frac{x - \sqrt{-2 + x^2}}{2t} \right\}, \left\{ u \rightarrow \frac{x + \sqrt{-2 + x^2}}{2t} \right\} \right\}$$

We store these solutions in the variables u1 and u2:

$$\{u1, u2\} = u /. L$$

$$\left\{ \frac{x - \sqrt{-2 + x^2}}{2t}, \frac{x + \sqrt{-2 + x^2}}{2t} \right\}$$

Beginning with u=u2, we draw at first one branch of the solution curve and then with u=u1 the second branch. Then all is shown in a single graphic:

$$u = u2$$

$$\frac{x + \sqrt{-2 + x^2}}{2t}$$

Inserting the first of the two solutions above gives a solution for $t \neq 0$

$$\text{Exp}[-(t u - x)^2]$$

$$e^{-\left(-x + \frac{1}{2} (x + \sqrt{-2 + x^2})\right)^2}$$

$$L2 = \text{Solve}[\text{Exp}[-(t u - x)^2] == u, t]$$

$$\left\{ \left\{ t \rightarrow \frac{1}{2} e^{\left(-x + \frac{1}{2} (x + \sqrt{-2 + x^2})\right)^2} (x + \sqrt{-2 + x^2}) \right\} \right\}$$

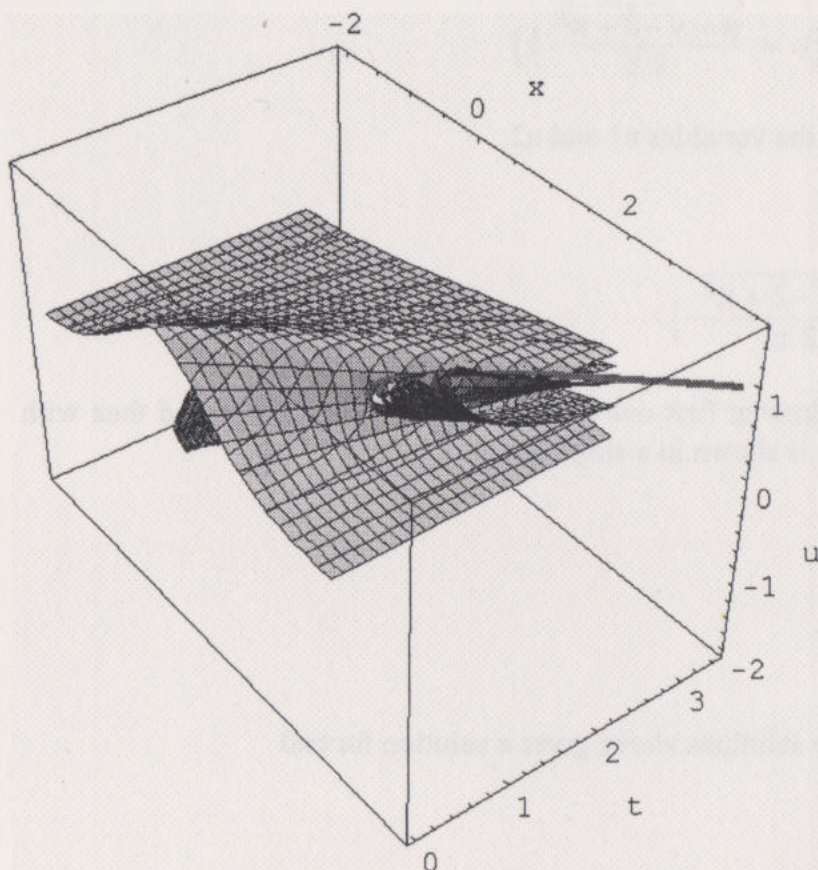
$$t = t /. \text{Flatten}[L2][[1]]$$

$$\frac{1}{2} e^{\left(-x + \frac{1}{2} (x + \sqrt{-2 + x^2})\right)^2} (x + \sqrt{-2 + x^2})$$

$$\text{Needs}["\text{Graphics`Master`}"]$$

$$K1 = \text{ParametricPlot3D}[\{x, t, u, \{\text{RGBColor}[1, 0, 0], \text{Thickness}[0.01]\}\}, \{x, \text{Sqrt}[2], 3.5\}, \text{DisplayFunction} \rightarrow \text{Identity};$$

$$\text{Show}[CP1, CP2, K1, \text{DisplayFunction} \rightarrow \$\text{DisplayFunction};$$



The projection of this branch of the curve onto the x-t plane is generated as a graphic, but not yet displayed:

```
cuspl=ParametricPlot3D[{x,t,0,{RGBColor[0,0,1],  

Thickness[0.01]}},{x,Sqrt[2],3.5},DisplayFunction->Identity];
```

Now we do the analogous calculation for the second branch

```
Clear[x,t,u]
```

```
u=u1
```

$$\frac{x - \sqrt{-2 + x^2}}{2t}$$

```
Exp[-(t u - x)^2]
```

$$E^{-\left(-x + \frac{1}{2} \left(x - \sqrt{-2 + x^2}\right)\right)^2}$$

```
L3=Solve[Exp[-(t u - x)^2]==u,t]
```

$$\left\{\left\{t \rightarrow -\frac{1}{2} E^{-\frac{1}{2} + \frac{1}{2} x \left(x + \sqrt{-2 + x^2}\right)} \left(-x + \sqrt{-2 + x^2}\right)\right\}\right\}$$

```
t=t/.Flatten[L3][[1]]
```

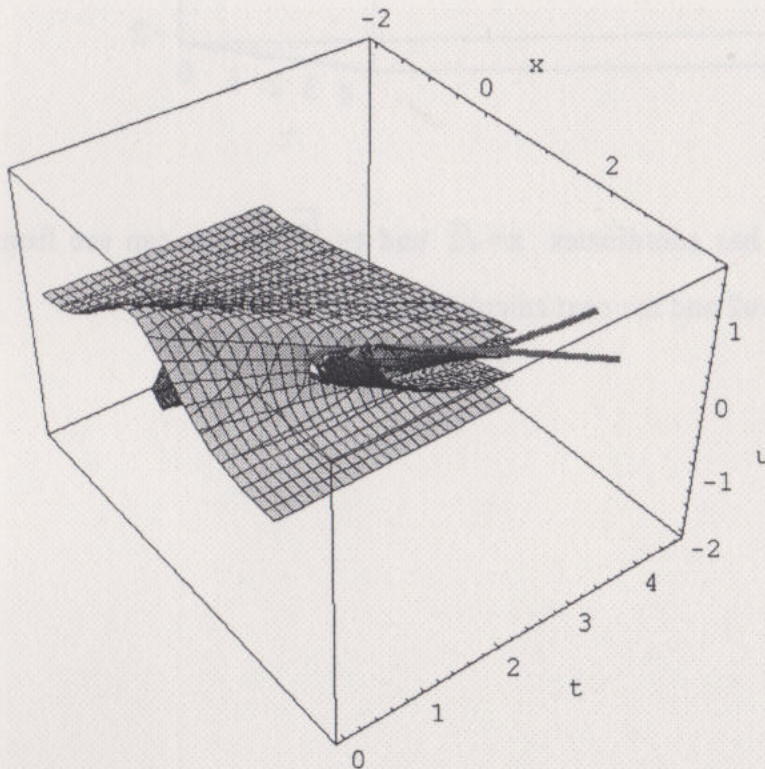

$$-\frac{1}{2} e^{-\frac{1}{2} + \frac{1}{2} x (x + \sqrt{-2 + x^2})} (-x + \sqrt{-2 + x^2})$$

```
K2=ParametricPlot3D[{x,t,u,{RGBColor[1,0,0],
  Thickness[0.01]}},{x,Sqrt[2],3.5},DisplayFunction->Identity];
```

```
cusp2=ParametricPlot3D[{x,t,0,{RGBColor[0,0,1],
  Thickness[0.01]}},{x,Sqrt[2],3.5},DisplayFunction->Identity];
```

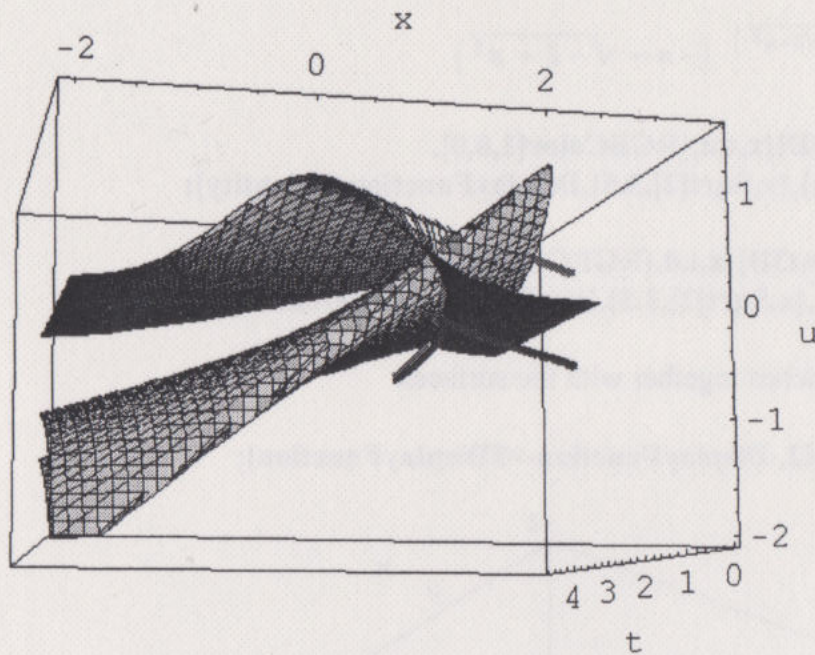
We show the two branches together with the surfaces

```
Show[CP1,CP2,K1,K2, DisplayFunction->$DisplayFunction];
```



Now the resulting graphic:

```
Show[CP1,CP2,K1,K2,cusp1, cusp2,
  ViewPoint->{-0.711, -3.074, -0.5},
  DisplayFunction->$DisplayFunction];
```



The cusp point itself has coordinates $x=\sqrt{2}$ und $t=\sqrt{\frac{e}{2}}$ as you can see from the formular above for $u=u_1$ and $u=u_2$ and the next calculation for t :

$$x=\text{Sqrt}[2];$$

t

$$\sqrt{\frac{e}{2}}$$

u

$$\frac{1}{\sqrt{e}}$$

$$\text{Punkt}=\{x,t,0\};$$

$$\text{Punkt3d}=\{x,t,u\};$$

We store the coordinates of the begin of the fold in three variables x_m , t_m , and u_m , since we shall use them later.

$$x_m=x;$$

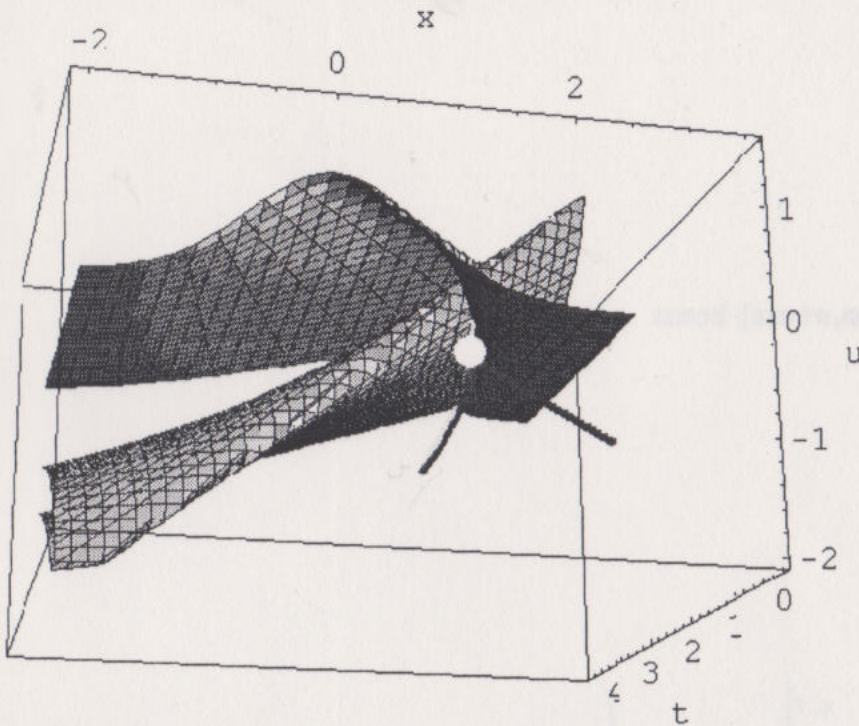
$$t_m=t;$$

$$u_m=u;$$

$$P=\text{Graphics3D}[\{\text{RGBColor}[0,1,0],\text{PointSize}[0.04],\text{Point}[\text{Punkt}]\}];$$

In the following graphic all objects, which we calculated until now, are shown from a viewpoint, where you can see the positions of the surface, the curves and the cusp in the x-t plane.

```
Show[CP1,CP2,K1,K2,cusp1, cusp2,P,  
ViewPoint->{-0.711, -3.074, -1.004},  
DisplayFunction->$DisplayFunction];
```



We clear the values of x,t,and u and compute some derivations of the function $F(x,t,u)$ by u in the starting point of the folding.

```
Clear[x,t,u]
```

```
D[F[x,t,u],u]/.{x->xm,t->tm,u->um}//Simplify
```

```
0
```

```
D[F[x,t,u],{u,2}]/.{x->xm,t->tm,u->um}//Simplify
```

```
0
```

```
D[F[x,t,u],{u,3}]/.{x->xm,t->tm,u->um}//Simplify
```

```
0
```

D[F[x,t,u],{u,4}]/.{x->x_m,t->t_m,u->u_m}//Simplify

$$\sqrt{2} E^{3/2}$$

As you can see, the first three derivatives in that point vanish, while the forth derivative is positive there. We now make such a parameter transformation of the x, t, u -space, that the origin of the new coordinate system is in the begin of the folding. Therefore we calculate $F(x+x_m, t+t_m, u+u_m)$ and subtract the constant $F(x_m, t_m, u_m)$. The transformed function is called F_t .

konst=F[x_m,t_m,u_m];

konst//N

0.819459

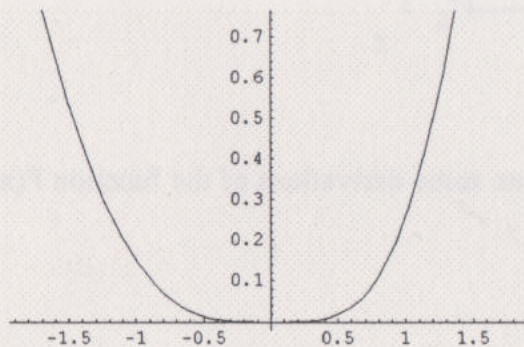
Ft[x_,t_,u_]:=F[x+x_m,t+t_m,u+u_m]-konst

We have $F_t(0,0,0)=0$:

Ft[0,0,0]

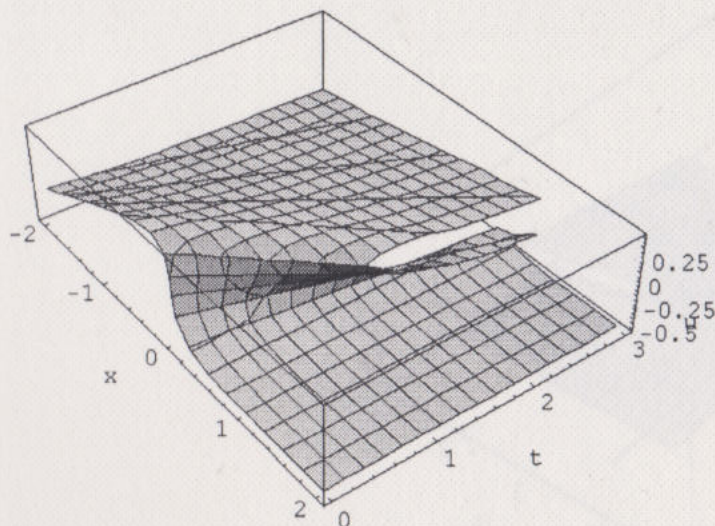
0

Plot[Ft[0,0,u],{u,-3,3}];



We also visualize the singular set of the new function:

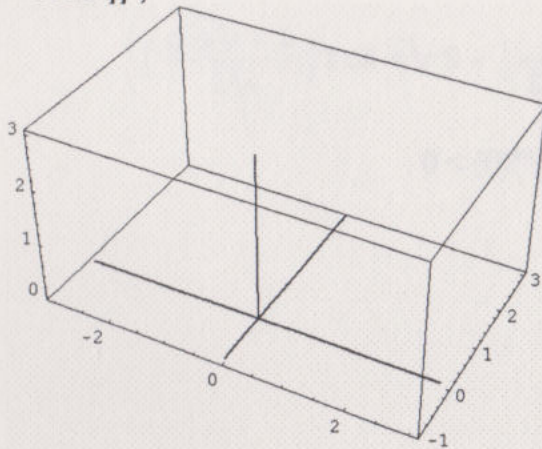
**CPG=ContourPlot3D[D[Ft[x,t,u],u],{x,-2,2},{t,0,3},{u,-2.5,2.5},
PlotPoints->{5,5,6},ViewPoint->{2,-1.5,2},Axes->True,
AxesLabel->{"x","t","u"}];**



We generate axes :

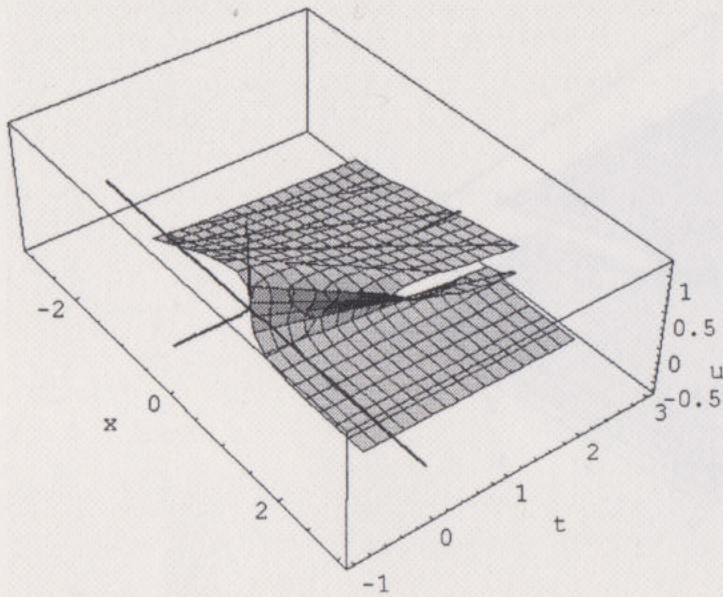
`Needs["Graphics`Master`"]`

```
ax=Show[ Graphics3D[ {{Thickness[0.005],
  RGBColor[0,0,1], Line[ {{3,0,0},{-3,0,0}} ],
  Line[ {{0,3,0},{0,-1,0}} ], Line[ {{0,0,3},{0,0,0}} ] }},
  Axes->True ] ;
```



We show the surface with axes

```
Show[CPG,ax];
```



The cusp (beginning of the folding) is now at the origin (0,0) of the parameter space.
Now we define function $g(u)$ by $Ft(0,0,u)$:

Remove[g]

$g[u_]=Ft[0,0,u]//Simplify$

$$\frac{1}{4} \left(2 \sqrt{2} u + \sqrt{2} e u^2 - 2 \sqrt{\pi} \operatorname{Erf}\left[\frac{1}{\sqrt{2}}\right] + 2 \sqrt{\pi} \operatorname{Erf}\left[\frac{1 - \sqrt{e} u}{\sqrt{2}}\right] \right)$$

We have: $g(0)=g'(0)=g''(0)=g'''(0)=0$, but $g''''(0) > 0$:

$g[0]$

0

$g'[0] //Simplify$

0

$g''[0] //Simplify$

0

$g'''[0] //Simplify$

0

$g''''[0] //Simplify$

$\sqrt{2} e^{3/2}$

Since $g''''[0] \neq 0$, this means $g \in \mathcal{m}(1)^4$. Thus there exists a coordinate transformation (diffeomorphism), such that in the new coordinates, which we again name u , the function g looks locally like the function u^4 . More about this transformation can be found in the book of Poston and Stewart: Catastrophe Theory., theorem 4.4, p.57. We have reached the point where catastrophe theory applies.

We think of g as a representative of a function germ and F_t as an unfolding of g with $p=(x,t)$ as unfolding parameter.

Since $\text{codim}(g) = \text{codim}(u^4) = \dim(\mathcal{m}(1)/\langle u^3 \rangle) = \dim(\langle u \rangle / \langle u^3 \rangle) = 2$, we get the cusp catastrophe as universal unfolding of g . There is a morphism of unfoldings of F_t to the cusp catastrophe: F_t is equivalent to the cusp catastrophe $F(x,t,u) = u^4 + x \cdot u^2 + t \cdot u$.

The reader will now probably ask, what this connection is good for. One reason for this approach to the problem, for example is the possibility, to carry over theorems of singularity theory about stability of unfoldings to the stability of the shock wave solutions of the pde (see Golubitsky, Schaeffer: "Stability of Shock Waves..", references). Moreover, this way, different phenomena in nature, e.g. caustics etc., can be described under a common aspect. Finally, we may use the much simpler Form of the cusp catastrophe, rather than the original function F , if we are interested in qualitative phenomena.

9 Short history of catastrophe theory

Catastrophe theory was founded by the French mathematician René Thom (born 2.9.1923). In the year 1972 Thom's famous book "Stabilité structurelle et morphogénèse" appeared. This book is a highly pretentious mixture of mathematics (application of deep differential topological theorems), philosophy and the applied natural sciences, especially biology. Thom had also great success on other mathematical themes. In 1958 he received the Fields-medal for his works in algebraic topology.

It was about 1930, when Marston Morse developed his theory of the singularities of differentiable functions (Morse theory). Hassler Whitney extended this theory in the 50th to singularities of differentiable maps of the plane into the plane.

John Mather (1960) introduced Algebra, especially the theory of ideals of differentiable functions, into singularity theory. Successors of Thom are:

Christopher Zeeman: He works on applications of catastrophe theory to physical systems.

Martin Golubitsky works with bifurcation theory, which is a succeeding theory to catastrophe theory.

John Guckenheimer worked with caustics and David Schaeffer published works on shock waves.

Gordon Wassermann wrote his doctoral thesis on "Stability of Unfoldings" (see references).

From the point of view of dynamical system the forerunners of Thom are: J. Henry ~~Poincaré~~ ^{Poincaré} (1854-1912) who looked at stability problems of dynamical systems, especially the problems of celestial mechanics.

Andrea Nikolaevic Kolmogorov, Vladimir I. Arnold and Jürgen Moser (KAM) influenced catastrophe theory with their works on stability problems (KAM-theorem).

10 Appendix: Introduction to Mathematica

You will find here a short summary of the Mathematica statements, that you need for the purposes of this book. But Mathematica is much more powerful. So it is worth to learn Mathematica thoroughly, for example from the book of Stephen Wolfram (s. reference).

Now let's assume, that you have installed Mathematica on your PC (version 4.0 or higher). Please, start the program. In the white window, you can do your input. In this book we use bold letters to distinguish between the lines that you give in, and the normally printed lines, that correspond to the answers of Mathematica.

Try some simple computations. Give in the following lines and finish each of it with the keys [Shift] and then [Return] (pressed together). The very first computation will need some more time, since the kernel of the system must be loaded at the beginning.

3+4.2*2

11.4

6-(0.342/3)^4 (The ^4 raises to the 4-th power)

5.999831103984

Mathematica can calculate exact:

3/4+5/6

$\frac{19}{12}$

The numerical (N) result of the last line (%) is

N[%]

1.5833333333333333

Mathematica can give you help: [Shift] and [F1] key.

Let's use some Mathematica functions like the square root (Sqrt) and the exponential function (Exp). Many more functions can be found in the Mathematica help system.

x=Sqrt[25]+Sin[Pi/4]+Exp[2]

$$5 + \frac{1}{\sqrt{2}} + e^2$$

x//N

13.096162880117198

// gives postfix form. of the operator N (numerical value)

Since, in the next example, the last term is a numerical expression (decimal point), the result of the complete calculation is given numerically.

x=Sqrt[25]+Sin[Pi/4]+Exp[2] Log[Abs[-3.4]]

14.74965209793543

To clear the value of the variable x, we write

Clear[x]

To remove x completely:

Remove[x]

To remove all user defined variables and functions, you may give in: Remove["Global`*"]

Now we define some new functions. Note the underline on the left side :

f[x_]:=Exp[Sin[x]]

f[4]

$e^{\sin[4]}$

The numerical value:

f[4]/N

0.469164

We need some operators: /. (Slash Point) uses a rule on its right side to apply it on the expression on its left side:

z=a+2

2+a

z^2 /. a->b

$$(2+b)^2$$

The operator /; makes it possible to formulate conditions: In the following example x gets the value 1 if $y > 0$ and x gets the value 0 if $y \leq 0$

```
x:=1/; y>0
x:=0/; y<=0
```

```
y=2;
```

```
x
```

```
1
```

```
y=-2;
```

```
x
```

```
0
```

We also need do-loops and if-alternatives

```
x=5;
y=7;
If[x>y, x, y]
```

```
7
```

Asking for equality:

```
If[x==y,x,y]
```

```
7
```

The do-loop in the next example prints the values from 1 to 3

```
Do[Print[i],{i,1,3}]
```

```
1
```

```
2
```

```
3
```

Two nested do-loops :

```
n=2;
Do[ Do[ Print[SequenceForm[i=i, k=k]] ,
      {i,1,n}], {k,1,n}]
```

$$i=1 \quad k=1$$

$$i=2 \quad k=1$$

$$i=1 \quad k=2$$

$$i=2 \quad k=2$$

Sums and products:

Sum[xⁱ,{i,1,9}]

$$x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9$$

Using in the main menu “Cell→ ConvertTo→ TraditionalForm“, after clicking on the input cell’s bracket, can change this input form into a formula. You may also use “palettes→ BasicInput”.

$$\sum_{i=1}^9 x^i$$

$$x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9$$

For products we have the following command (for details always look at the system help):

Product[i²,{i,1,5}]

$$14400$$

or

$$\prod_{k=1}^5 k^2$$

$$14400$$

Sum[i², {i, 1, n}]

$$\frac{1}{6} n (1 + n) (1 + 2 n)$$

This can now be converted by „TraditionalForm“ into

$$\sum_{i=1}^n i^2$$

$$\frac{1}{6} n (n + 1) (2 n + 1)$$

Now we do some algebraic transformations:

Expand[(a+b)^3]

$$a^3 + 3a^2b + 3ab^2 + b^3$$

Factor[a^2 - 2a b + b^2]

$$(a - b)^2$$

Cancel[(x+1)(x^2-2x+1)/(x-1)]

$$(-1+x)(1+x)$$

u=3/x+5/y

$$\frac{3}{x} + \frac{5}{y}$$

Together[u]

$$\frac{5x + 3y}{xy}$$

Apart[%]

$$\frac{3}{x} + \frac{5}{y}$$

The Apart command must not be confused with the Part command:

Part[u,1]

$$\frac{3}{x}$$

This takes the first part of the term u. You can also do this by double brackets (shown here for the second part of the term u):

u[[2]]

$$\frac{5}{y}$$

In order to simplify expressions, you may do as follows:

expr=7*((x-a)^4-2(x-a)^3)/(x-a)^2;

Simplify[expr]

$$7(a-x)(2+a-x)$$

Here are some examples for trigonometric formulas. The Greek letter alpha is generated with the "ESC-key" then "a" and "ESC-key".

TrigExpand[Sin[α + β]]

$$\cos[\beta] \sin[\alpha] + \cos[\alpha] \sin[\beta]$$

TrigExpand[Sin[2 α]]

$$2 \cos[\alpha] \sin[\alpha]$$

TrigReduce[Sin[α]*Cos[α]]

$$\frac{1}{2} \sin[2\alpha]$$

In order to treat several objects as a single one, you can use lists. In the following lines a list, named y, has 5 elements. The elements are raised to the second power here, and the third element is gripped out.

y={3, 1, 7, 4.1, -6.2}

{3, 1, 7, 4.1, -6.2}

y^2

{9, 1, 49, 16.81, 38.440000000000005}

y[[3]]

49

To remove parentheses in lists of list, we have the Flatten command:

u={{1,3,5},{2},{1,4}};

v=Flatten[u]

{1, 3, 5, 2, 1, 4}

Lists are often generated by the Table-command:

liste=Table[i^3, {i, 1, 10}]

{1, 8, 27, 64, 125, 216, 343, 512, 729, 1000}

This generates a list, which contains the third power of the numbers 1 to 10

Table[{i,i^2},{i,1,5}]]//TableForm

```
1  1
2  4
3  9
4 16
5 25
```

Comments can be made between (* and *). The Apply-command is often used together with the operator for sums or the Times-command for products:

(* this is adding rest. multiplying elements of a list named list1*)

list1={2, 5, 1, 7};

Apply[Plus, list1]

Apply[Times, list1]

```
15
70
```

Union gives the union set of lists:

list1={1,2,4,7} ;

list2={3,2,5,6};

Union[list1, list2]

```
{1,2, 3, 4, 5, 6, 7}
```

Here are two other useful commands:

Append[{2,4,9,7},9]

```
{2, 4, 9,7, 9}
```

Max[list1]

```
7
```

Lists can be used to define vectors:

v={2,5,1,8}

```
{2,5,1,8}
```

This list is also a 4 dimensional vector. If you want to see it as a column vector:

ColumnForm[v]

2
5
1
8

v can be multiplied by real numbers:

c=3;
c v

{6, 15, 3, 24}

Length[v]

4

(Length is not its geometrical length, but the number of components of v). Vectors, which have the same number of components, can be added:

u={3,-1,5,6}

{3, -1, 5, 6}

w=u+v

{5, 4, 6, 14}

r=u-v

{1, -6, 4, -2}

The dot product:

s=u.v

54

A matrix is a list of vectors of equal number of components. The matrix M:

M={{1,5},{-4,2}};
M//MatrixForm

$$\begin{pmatrix} 1 & 5 \\ -4 & 2 \end{pmatrix}$$

Its transposed matrix:

T=Transpose[M];
T//MatrixForm

$$\begin{pmatrix} 1 & -4 \\ 5 & 2 \end{pmatrix}$$

If you want to distinguish between row vectors and column vectors, you can do this as follows. A row vector:

v={{2,7,-1}};

v//MatrixForm

$$(2 \ 7 \ -1)$$

A column vector:

w = {{-1}, {3}, {5}};

w // MatrixForm

$$\begin{pmatrix} -1 \\ 3 \\ 5 \end{pmatrix}$$

Certain elements, rows or columns of a matrix can be gripped out. Define a Matrix:

P={{2,3,4},{1,3,6}};

P//MatrixForm

$$\begin{pmatrix} 2 & 3 & 4 \\ 1 & 3 & 6 \end{pmatrix}$$

This gives the element $P_{2,3}$, i.e. the element in the second row and in the third column:

P[[2,3]]

6

This gives the second row:

P[[2]]

{1, 3, 6}

and the second column

Transpose[P][[2]]

{3, 3}

If you have got data in the following form

dat={2,7,1,3,8,-1,4,9,0};

you can create a matrix from the data, e.g. a matrix with 3 columns:

B=Partition[dat,3];

B//MatrixForm

$$\begin{pmatrix} 2 & 7 & 1 \\ 3 & 8 & -1 \\ 4 & 9 & 0 \end{pmatrix}$$

To commute the rows of this matrix, you can give in:

{ B[[1]], B[[2]] } = { B[[2]], B[[1]] }

{{3, 8, -1}, {2, 7, 1}}

B//MatrixForm

$$\begin{pmatrix} 3 & 8 & -1 \\ 2 & 7 & 1 \\ 4 & 9 & 0 \end{pmatrix}$$

To commute its columns, you must transpose the matrix, then commute its rows and then transpose it again.

If you have two lists of equal length

a={2,3,6};

b={4,8,9};

and you want to treat them as coordinates of points, you may do the following:

c=Transpose[{a,b}]

{{2, 4}, {3, 8}, {6, 9}}

With Mathematica you can easily do the product of two matrices

T.M//MatrixForm

$$\begin{pmatrix} 17 & -3 \\ -3 & 29 \end{pmatrix}$$

But please be careful: If you write $T * M$, you will get a multiplication of corresponding matrix elements, but not the usual matrix product, as you know it from mathematics.

T * M //MatrixForm

$$\begin{pmatrix} 1 & -20 \\ -20 & 4 \end{pmatrix}$$

You can compute T/M (a division of corresponding elements). This makes sense for lists but not for matrices. If you write A^3 , this is not the third matrix power, as you will get it, if you write $A.A.A$ or equivalently $\text{MatrixPower}[A,3]$. The expression A^3 means $A*A*A$, where each component is raised to the third power.

M={{1,5},{-4,2}};

MatrixPower[M,3]//MatrixForm

$$\begin{pmatrix} -79 & -65 \\ 52 & -92 \end{pmatrix}$$

Special matrices are:

IdentityMatrix[4]//MatrixForm

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A=DiagonalMatrix[{2,7,3,8}];

A//MatrixForm

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 8 \end{pmatrix}$$

H=DiagonalMatrix[{2,-1}];

H//MatrixForm

$$\begin{pmatrix} 2 & 0 \\ 0 & -1 \end{pmatrix}$$

General matrices:

Clear[x]

Q=Array[x,{2,3}];

Q//MatrixForm

$$\begin{pmatrix} x[1, 1] & x[1, 2] & x[1, 3] \\ x[2, 1] & x[2, 2] & x[2, 3] \end{pmatrix}$$

Its number of rows and columns:

Dimensions[Q]

{2, 3}

You may also use the table command:

Table[a[i,j], {i,1,3}, {j,1,2}] //MatrixForm

$$\begin{pmatrix} a[1, 1] & a[1, 2] \\ a[2, 1] & a[2, 2] \\ a[3, 1] & a[3, 2] \end{pmatrix}$$

The inverse of a quadratic matrix

M={{3,4},{-4,6}};

Inverse[M]//MatrixForm

$$\begin{pmatrix} \frac{3}{17} & -\frac{2}{17} \\ \frac{2}{17} & \frac{3}{34} \end{pmatrix}$$

Its determinant:

Det [M]

34

Next we try “pure functions”: Instead of writing

f[x_]:=x^2

Map[f,{2,4,a,b,7}]

{4, 16, a², b², 49}

you can use a pure function, with the symbols # and &

Map[#^2&,{2,4,a,b,7}]

{4, 16, a², b², 49}

Solving equation in Mathematica is made like this

g=Solve[4x-2==0,x]

$$\left\{\left\{x \rightarrow \frac{1}{2}\right\}\right\}$$

To have the value of x without parentheses write:

x=x/.Flatten[g]

$$\frac{1}{2}$$

Test it:

3*x

$$\frac{3}{2}$$

Next, we solve a quadratic equation:

Remove[x]

eq=Solve [x^2-3 x+ 5==0,x]

$$\left\{\left\{x \rightarrow \frac{1}{2} (3-i \sqrt{11})\right\}, \left\{x \rightarrow \frac{1}{2} (3+i \sqrt{11})\right\}\right\}$$

In order to get the solutions without parenthesis, you give in:

{x1,x2}=x/.eq

$$\left\{\frac{1}{2} (3-i \sqrt{11}), \frac{1}{2} (3+i \sqrt{11})\right\}$$

For example test x1:

x1

$$\frac{1}{2} (3-i \sqrt{11})$$

Let's now differentiate and integrate functions:

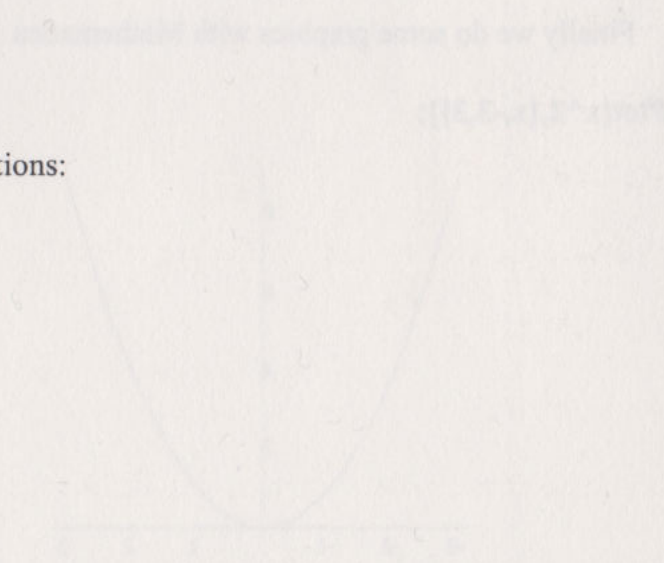
f[x_]:=x^2

f'[x]

2 x

or equivalently

D[f[x],x]



2 x

A function in two variables:

f[x_,y_]:=x^2+x*y^3

Its partial derivative with respect to y:

D[f[x,y],y]

3 x y^2

Second partial derivative with respect to y

D[f[x,y],{y,2}]

6 x y

Integration:

Integrate[x^n,x]

$$\frac{x^{1+n}}{1+n}$$

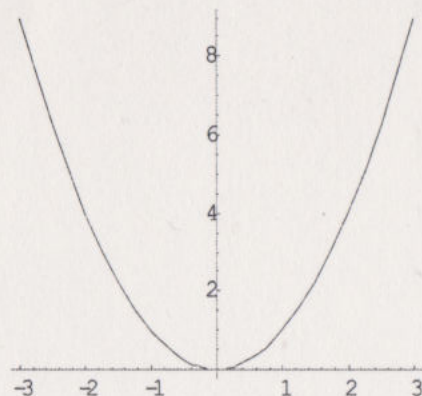
Integration from 1 to 2:

Integrate[x^3,{x,1,2}]

15/4

Finally we do some graphics with Mathematica

Plot[x^2,{x,-3,3}];



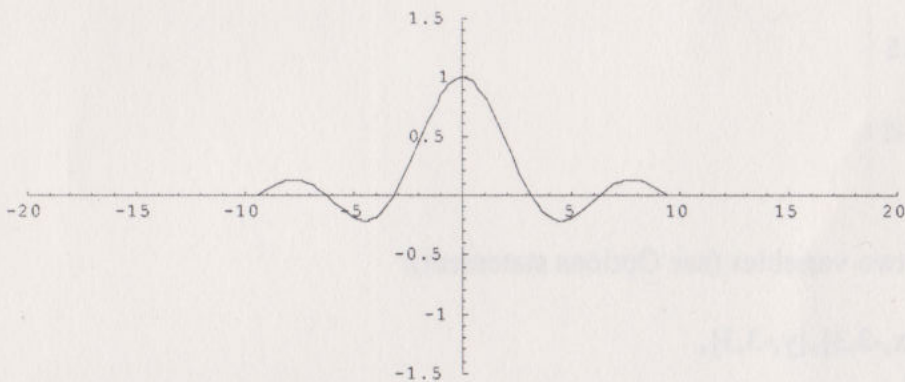
The “Options” command is used to see all possibilities of the Plot statement:

Options[Plot]

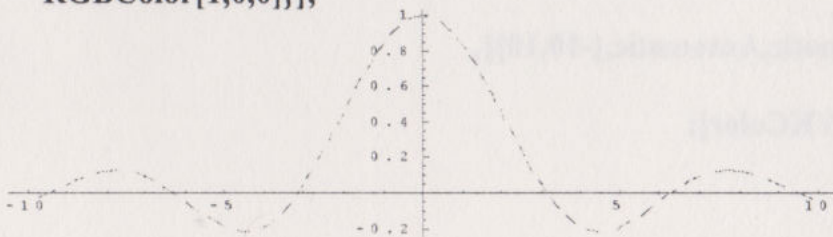
```
{AspectRatio→ $\frac{1}{\text{GoldenRatio}}$ , Axes→Automatic,
AxesLabel→None, AxesOrigin→Automatic,
AxesStyle→Automatic, Background→Automatic,
ColorOutput→Automatic, Compiled→True,
DefaultColor→Automatic, Epilog→{}, Frame→False,
FrameLabel→None, FrameStyle→Automatic,
FrameTicks→Automatic, GridLines→None,
ImageSize→Automatic, MaxBend→10.,
PlotDivision→30., PlotLabel→None, PlotPoints→25,
PlotRange→Automatic, PlotRegion→Automatic,
PlotStyle→Automatic, Prolog→{}, RotateLabel→True,
Ticks→Automatic, DefaultFont→$DefaultFont,
DisplayFunction→$DisplayFunction,
FormatType→$FormatType, TextStyle→$TextStyle}
```

Examples:

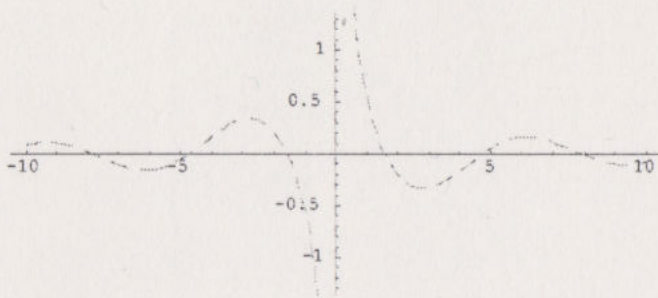
Plot[Sin[x]/x,{x,-3 Pi,3 Pi},PlotRange->{{-20,20},{-1.5,1.5}}]



**P=Plot[Sin[x]/x,{x,-10,10},PlotStyle->{Dashing[{0.03,0.02}],
RGBColor[1,0,0]}];**

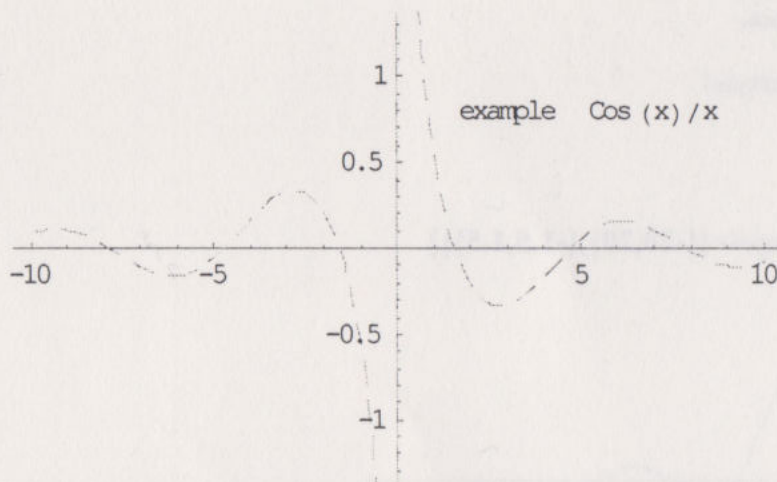


**P2=Plot[Cos[x]/x,{x,-10,10},PlotStyle->{Dashing[{0.03,0.02}],
RGBColor[1,0,0]}];**



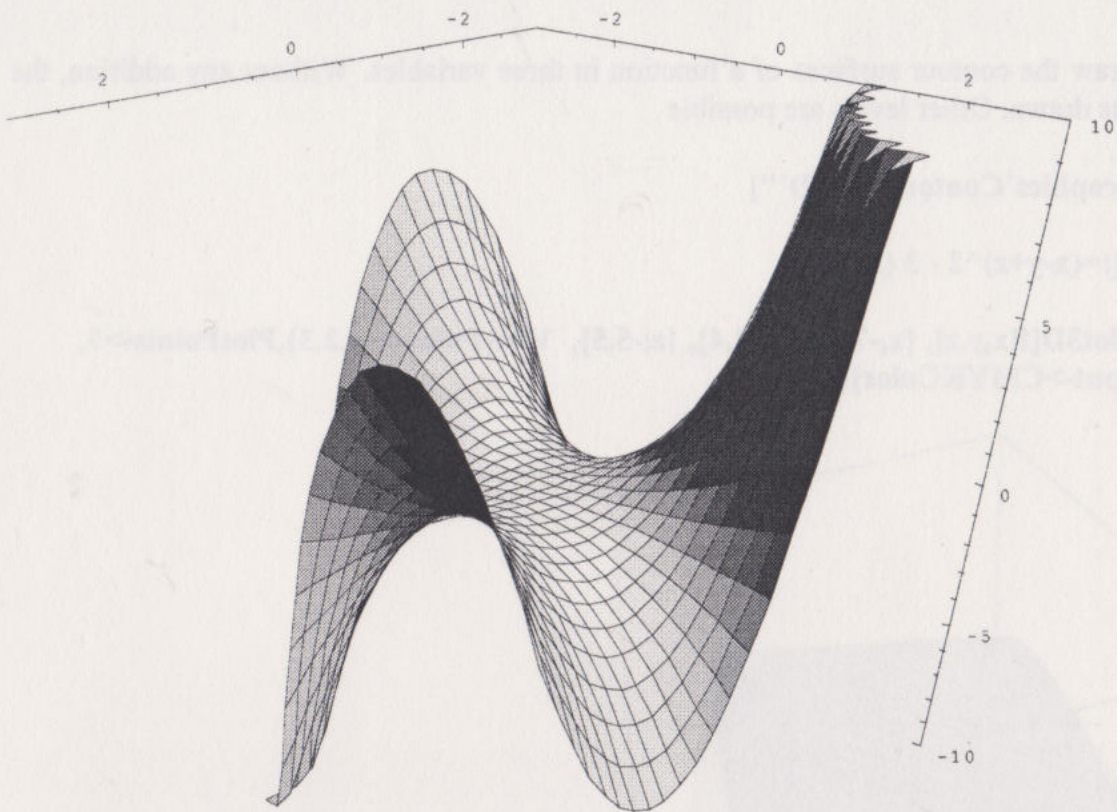
```
T=Graphics[Text[" example Cos(x)/x",{5,0.8}]];
```

```
Show[P2,T];
```



The graph of a function in two variables (see Options statement):

```
B=Plot3D[x^2*y-y^2*x,{x,-3,3},{y,-3,3},  
  PlotPoints->30,  
  ViewPoint->{1,1,1},  
  Boxed->False,  
  BoxRatios->{1,1,2},  
  PlotRange->{Automatic,Automatic,{-10,10}},  
  ClipFill->None,  
  ColorOutput->CMYKColor];
```

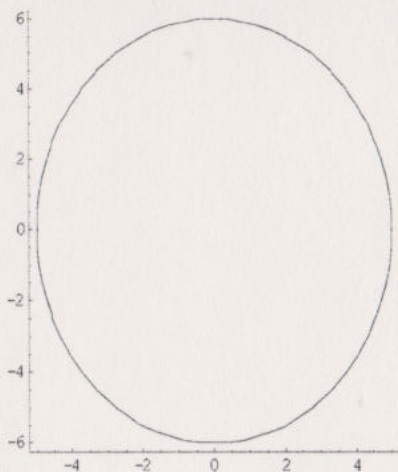
Please note: A method to smoothen out the border is shown in the first chapter (example: pig trough).

Next we wish to draw an implicitly given curve:

```
Remove["Global`*"]
```

```
Needs["Graphics`Master`"]
```

```
ImplicitPlot[x^2/25+y^2/36==1,{x,-5,5},{y,-6,6}];
```

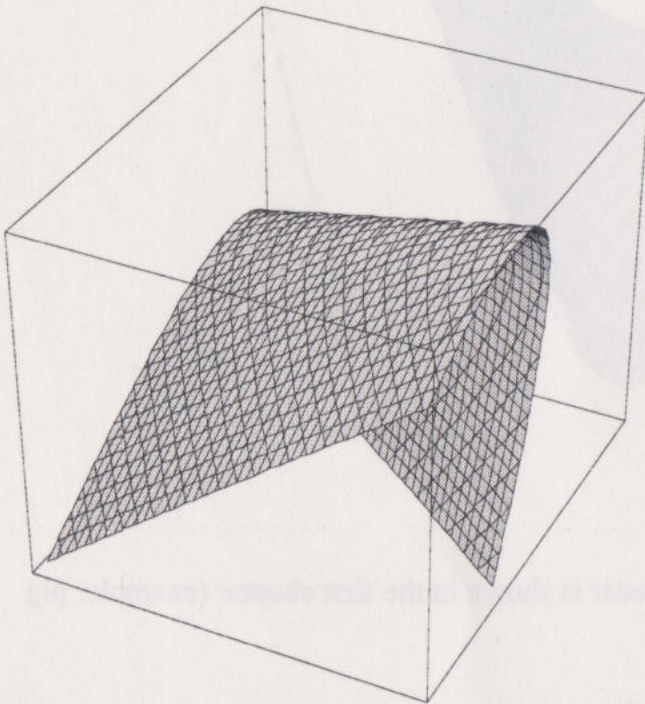


Next we draw the contour surfaces of a function in three variables. Without any addition, the zero level is drawn. Other levels are possible

Needs["Graphics`ContourPlot3D`"]

$f[x,y,z] := (x-y+z)^2 - 3(x-z)$

ContourPlot3D[f[x,y,z], {x,-7,7}, {y,-4,4}, {z,-5,5}, ViewPoint->{4,2,3}, PlotPoints->7,
ColorOutput->CMYKColor];



Please note: The plotting time depends strongly on the number of "PlotPoints" you choose !

*

11 Index

#
 $\varepsilon(n)$ · 38 $\varepsilon_p(n,k)$ · 38

%

% · 149

/

/ · 151

/. · 150

// · 150

[

[[· 153, 157

^

^ · 159

>

-> · 150

A

algebra · 38

Apart · 153

Append · 155

appendix · 149

Apply · 155

Arnold · 148

Array · 159

B

bifurcation set · 71

butterfly catastrophe · 106

C

C^∞ · 38
Cancel · 153
catastrophe set · 68, 95
catastrophe surface · 81
chain rule · 18
characteristics · 130
classification theorem · 48, 58
Clear · 150
 $\text{codim}(f)$ · 57
codimension · 49, 56, 58, 92, 106, 114, 120, 123
ColumnForm · 155
conservation law · 126
ContourPlot3D · 34, 67, 90, 100, 107, 136
control space · 57
ConvertTo · 127
coordinate transformation · 56
cubic · 56
curve · 13, 16, 17, 18, 19, 20, 21, 22, 63, 66, 71, 73, 74, 75, 76, 80, 83, 84, 86, 87, 95, 96, 109, 127, 130, 131, 135, 138, 139, 140
cusp · 17, 71, 142, 143, 146

D

degenerated · 27, 49, 68
derivative · 16
determinacy · 47
determinacy test · 47
determinant · 31, 46, 124, 130, 131
DiagonalMatrix · 159
diffeomorphism · 31, 33, 34, 43, 46, 48
differentiable · 26, 38
Dimensions · 160
directional derivative · 18, 21
disturbance · 50, 53
Do · 151
Dynamic-Visualizer · 113

E

eigenvalue · 46
elliptical umbilic · 120
endogenous Variable · 57
entropy condition · 126
equivalence relation · 37
exogenous variable · 57
Expand · 153
extrema · 29, 66

F

Factor · 153
families · 29
family · 51
fibers · 57
fold catastrophe · 62
fold line · 69
function · 18

G

generated · 38, 42
generating vector · 17
germ · 37
Golubitsky · 148
gradient · 18
graph · 18
Gromoll-Meyer lemma · 49
Guckenheimer · 148

H

Hessian matrix · 27, 46, 47, 48, 49, 115, 120, 124
history · 148
hysteresis · 84

I

ideal · 38
IdentityMatrix · 159
If · 151
immersion · 48
initial value problem · 126
integral surface · 136
interval · 16
inverse map · 31

J

Jacobian ideal · 56
Jacobian matrix · 26, 31, 46, 48, 130
jump condition · 126

K

k-determined · 47
k-jet · 39

Kolmogorov · 148

L

Length · 156

line · 16

linear combination · 24

linearly independent · 23

ListPlotVectorField3D · 14

M

manifold · 33

map · 18, 26

Map · 160

map germ · 37

Mather · 148

MathLive · 113

MatrixForm · 156, 160

MatrixPower · 159

Max · 155

Maxwell convention · 84

Maxwell set · 87

morphism · 56

Morse · 148

Morse form · 48

Morse lemma · 48

Moser · 148

multiindex · 39

N

N · 149

normal form · 58

normal vector · 126

O

organization center · 58

Outer · 20

P

parabola · 16

parabolic umbilic · 123

parameter curve · 16

parameter space · 21, 24, 71, 73, 96, 104, 106, 116, 131

parameterization · 127

parameterized surface · 23
 ParametricPlot3D · 13, 21, 34, 36, 82, 83, 86, 95, 97, 101, 103, 105, 121, 139, 140, 141
 Part · 153
 Partition · 158
 perfect delay rule · 84
 pig trough · 29
 Pincer · 148
 Plus · 155
 point · 24
 polynomial · 54
 postfix form · 150
 potential · 29
 Print · 151
 product ideal · 42
 projection · 21

Q

qualitative · 44
 quasilinear · 127

R

radius · 34
 range · 43
 rank · 27
 region · 23
 regular · 17, 27, 48
 Remove · 46
 representative · 38
 r-equivalent · 43, 47
 Riemann-Hugoniot catastrophe · 127
 right-equivalent · 43
 right-left-equivalence · 43
 Ring · 38

S

saddle · 47
 Schaeffer · 148
 SequenceForm · 151
 Shift · 149
 shock curves · 126
 shock wave · 126
 Siersma's trick · 58
 Simplify · 153
 singular · 16, 17, 27, 29, 48
 singularity · 27

singularity surface · 81
singularity theory · 48
SpinShow · 112
splitting lemma · 49
stable · 57
state space · 57
structural instability · 49
structural unstable · 53
submersion · 48
sum · 42
surface · 19, 23, 24, 34, 66, 84, 108
system of equations · 94

T

Table · 154, 155, 160
TableForm · 155
tangent · 16, 17, 21
tangent plane · 24
tangent space · 127
tangent vector · 18
Taylor · 39, 42
Thom · 148
total differential · 26
 $T_p S$ · 127
TraditionalForm · 127
Transpose · 156
Tschirnhaus-transformation · 56

U

unfolding · 53
unfolding dimension · 53
unfolding morphism · 57
Union · 155
universal · 56

V

vector field · 127
vector space · 38
versal · 57

W

Wassermann · 148
Whitney · 148

12 References

Arnold, V.I.	Catastrophe Theory, Springer ,1984
Bröcker, Th.	Analysis, Bd. 1-3, Spektrum Akad. Vlg. ,(1995)
Bröcker, Th., Jänich, K.	Einführung in die Differentialtopologie, Heidelberger Taschenbücher , 1973 und 1990
Bröcker,Th.	Differentiable Germs and Catastrophes, London Mathem. Soc. Lecture Notes Series, Cambridge Univ.Press
Bruce, J.W., Gibblin, P.J.	Curves and Singularities, Cambridge Univ. Press, Second Edit. 1992
Castrigiano, D.,; Hayes, S.A.	Catastrophe Theory, Addison Wesley, 1993
Chilligworth,D. R.J.	Differential topology with a view to applications, Pitman ,1976
Forster, O.	Analysis (Bd. 1-3), Vieweg
Gilmore,R.	Catastrophe Theory for Scientists and Engineers, Wiley, 1981
Golubistky,M.	An Introduction to Catastrophe Theory, Dep.of Mathem.,Queens College, Flushing, NY 11367
Golubitsky, M.; Schaeffer, D.G.	Stabilty of Shock Waves for a Single Conservation Law, Adv. in math. 16 (1975)
Golubitsky, M.; Schaeffer, D.G.	Singularities and Groups in Bifurcation Theory, Springer,1985
Golubitsky,M., Guillemin,V.	Stable Mappingss and their Singularities, Springer, G.T.M. 14
Guckenheimer,J.	Solving a Single Conservation Law, Springer, Lecture Notes 468
Guckenheimer,J.	Catastrophes and Partial differential Equations, Ann. Inst. Fourier Grenoble Vol.23,1973,S. 31-59
Lax, P.	The Formation and Decay of Shock waves, Amer. Math. Monthly, March (1972), p.227-241
Lu	Singularity Theory with an Introduction to CatastropheTheory, Springer, 1976
Majthay,A.	Foundations of Catastrophe Theory, Pitman,1985
Mather, J.N.	Stability of C^∞ -Mappings I , Annals of math,Vol. 87,89-104,II: Annals of math, Vol. 89/2,254-291
Poston,T. Stewart,J.	Taylor Expansions and Catastrophes, Pitman, 1976
Poston. T. Stewart,J.	Catastrophe Theory and its Applications, Pitman, 1978
Sanns, W.	Schockwellen und Katastrophentheorie, Thesis Univ. Heidelberg, 1977. In German
Sanns, W. ; Schuchmann, M.	Mathematik für Wirtschaftswissenschaftler und Ingenieure mit Mathematica (in German),Oldenbourg, 1999. (This is an introduction to Mathematica and Engineering Mathematics).
Saunders.P.T.	Katastrophentheorie, Vieweg, 1986
Thom, R.	Structural Stability and Morphogenesis, Benjamin Inc. (1975), Reading,

	Massachusetts
Ursprung, H.W.	Die elementare Katastrophentheorie: Eine Darstellung aus der Sicht der Ökonomie, Lecture Notes in Economics and Mathematical Systems, Springer
Wassermann, G.	Stability of Unfoldings, Lecture Notes in Math. Nr 393, Springer (1974)

ISB N 3-934366-76-7

DEM 39.80